

# Inverse Modeling of Dynamical Systems: Multi-Dimensional Extensions of a Stochastic Switching Problem\*

Erik Bates<sup>†</sup>      Blake Chamberlain<sup>‡</sup>      Rachel Gettinger<sup>§</sup>

August 20, 2012

## Abstract

The Buridan's ass paradox is characterized by perpetual indecision between two states, which are never attained. When this problem is formulated as a dynamical system, indecision is modeled by a discrete-state Markov process determined by the system's unknown parameters. Interest lies in estimating these parameters from a limited number of observations. We compare estimation methods and examine how well each can be generalized to multi-dimensional extensions of this system. By quantifying statistics such as mean, variance, frequency, and cumulative power, we construct both method of moments type estimators and likelihood-based estimators. We show, however, why these techniques become intractable in higher dimensions, and thus develop a geometric approach to reveal the parameters underlying the Markov process. We also examine the robustness of this method to the presence of noise.

---

\*This research is supported in part by NSA grant H98230-11-1-0222 and NSF grant DMS-1062817.

<sup>†</sup>Michigan State University, [bateser2@msu.edu](mailto:bateser2@msu.edu)

<sup>‡</sup>Susquehanna University, [chamberlainb@susqu.edu](mailto:chamberlainb@susqu.edu)

<sup>§</sup>Saint Vincent College, [rachel.gettinger@email.stvincent.edu](mailto:rachel.gettinger@email.stvincent.edu)

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Donkey on a Line</b>	<b>3</b>
2.1	Method of Moments Type Estimators . . . . .	4
2.1.1	The Markov Process . . . . .	4
2.1.2	Continuous Dynamics . . . . .	6
2.1.3	Cumulative Power . . . . .	8
2.1.4	Couplet Inversion and Results . . . . .	11
2.2	Likelihood-Based Estimation . . . . .	12
2.3	State Detection . . . . .	14
<b>3</b>	<b>Donkey in a Triangle</b>	<b>15</b>
3.1	The Markov Process . . . . .	15
3.2	Continuous Dynamics . . . . .	16
3.3	State Detection . . . . .	18
<b>4</b>	<b>Noisy Data</b>	<b>19</b>
4.1	Linear Regression . . . . .	20
4.2	Denoising Techniques . . . . .	20
4.2.1	Locally Weighted Polynomial Regression . . . . .	21
4.2.2	Wavelet Filtering . . . . .	22
4.2.3	Butterworth Filtering . . . . .	22
4.2.4	Total Variation Denoising . . . . .	24
<b>5</b>	<b>Conclusion</b>	<b>26</b>
	<b>Appendices</b>	<b>28</b>
A	An Alternative Perspective . . . . .	28
A.1	Expected Waiting Times . . . . .	29
A.2	Transition Probabilities . . . . .	30
A.3	Parameter Estimation . . . . .	31
B	Eigenvector of Markov Matrix with Eigenvalue 1 . . . . .	32
C	A Conjecture Regarding Eigenvectors . . . . .	35
D	The Growth of Cumulative Power . . . . .	38
	<b>Acknowledgments</b>	<b>38</b>
	<b>References</b>	<b>38</b>

# 1 Introduction

In the classical paradox known as “Buridan’s Ass,” first posed in the 14th century by Jean Buridan, an ass is placed midway between a pail of water and a bale of hay. Assuming the donkey prefers the closer object, he theoretically never decides to approach one or the other when placed equidistant between the two (Lamport, 2012). This problem can be generalized as a dynamical system in which the donkey is free to roam between the two objects, nevertheless changing his preference before ever actually reaching one or the other. In this formulation, the donkey’s indecision is modeled by a discrete-state Markov process, the state being the donkey’s current preference. The parameters governing this Markov process are the probabilities of the donkey switching that preference. We begin by exploring the mathematical generalization of the classical, two-state paradox with the goal of estimating these parameters. We consider both method of moments type estimators and likelihood-based estimators to achieve this goal. We also develop an algorithm that allows us to directly estimate the parameters.

We are interested in the behavior of the system in higher dimensions, so we also consider the donkey in a triangular pen. By adding one more state to the system and extending the problem from one to two dimensions, many of the methods we explore for the donkey on a line become intractable. Nevertheless, our algorithm, which detects the donkey’s state at each unit time step, still allows us to measure the parameters of the system. In the absence of measurement noise, this state detector gives us the most reasonable values for our parameters. In the presence of noise, though, the state detector performs poorly. Consequently, we employ methods of denoising the system before applying state detection.

## 2 Donkey on a Line

Following the setup of Buridan’s Ass, we place the donkey on a line between a pail of water at  $x = 0$  and a bale of hay at  $x = 1$ . We define state 0 as the state in which the donkey is moving towards the pail of water, and state 1 as the state in which the donkey is moving towards the bale of hay.<sup>1</sup> The donkey slows as he comes closer to either object, so he never actually reaches the water or the hay. The donkey’s movement is modeled by the following differential equations:

$$\text{State 0: } \frac{dx}{dt} = v(-x) \tag{1}$$

$$\text{State 1: } \frac{dx}{dt} = v(1 - x). \tag{2}$$

The constant  $v$  controls the speed of the donkey. While the donkey obeys these differential equations in a continuous fashion, we allow him to change state only at unit time steps. We

---

<sup>1</sup>For simplicity, the donkey is always initially placed at  $x = 0.5$  and is taken to be in state 0.

let  $\tau_{01}$  be the probability that the donkey switches from state 0 to state 1 at any given time step, and  $\tau_{10}$  be the probability that the donkey switches from state 1 to state 0. These are the parameters we estimate in our mathematical model of Buridan’s Ass. We assume we know the constant  $v$  (even if not known, it is easily measured), and we observe only the donkey’s position at each of finitely many time steps. Notice that the parameters  $\tau_{01}$  and  $\tau_{10}$  change with neither time nor the donkey’s current state, imparting them the independence that characterizes the stochastic switching of this system as a Markov process.

## 2.1 Method of Moments Type Estimators

One way to estimate  $\tau_{01}$  and  $\tau_{10}$  is the method of moments approach. The standard method of moments estimation scheme uses moments of measured data, such as mean and variance, to estimate unknown parameters (Hansen, 1982). The approach relies on deriving algebraic expressions for these moments in terms of the parameters. When these expressions are invertible, they give closed-form estimates of parameters in terms of only measured data. Furthermore, this technique can be used with feature statistics such as frequency and cumulative power. Since the donkey on a line system has two parameters, we require couplets, or pairs of invertible expressions, to estimate  $\tau_{01}$  and  $\tau_{10}$ .

### 2.1.1 The Markov Process

The probability distribution of the donkey’s state at the  $n$ th time step is given by the discrete-state Markov chain

$$p_n = Ap_{n-1}, \quad (3)$$

where

$$A = \begin{bmatrix} 1 - \tau_{01} & \tau_{10} \\ \tau_{01} & 1 - \tau_{10} \end{bmatrix} \quad (4)$$

and  $p_0$  is a vector whose entries are the initial probabilities of the donkey being in either state. Notice that  $A$  is a column stochastic matrix (one whose columns each have entries summing to 1). Any such matrix has an eigenvector associated with an eigenvalue of 1 (see Appendix B). It is a well-known fact that under certain conditions<sup>2</sup>, this eigenvector is unique<sup>3</sup> and represents a stable distribution to which any initial distribution vector will

---

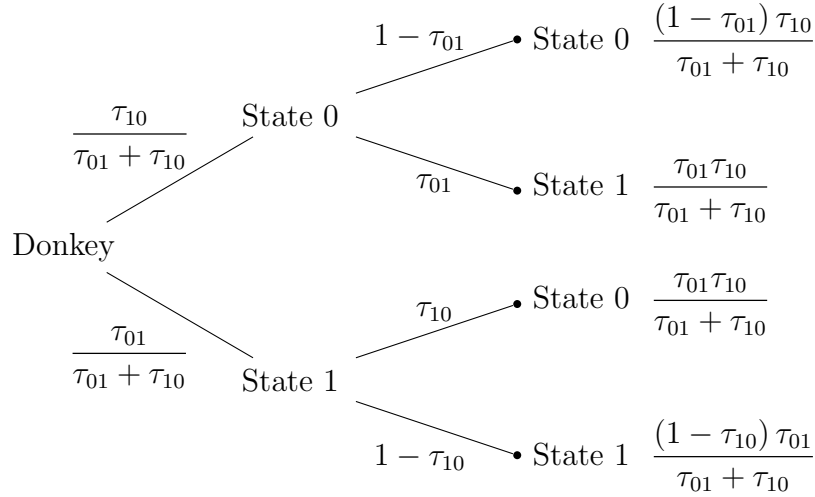
<sup>2</sup>The Markov chain must be recurrent, aperiodic, and irreducible (Orey, 1962). A recurrent Markov chain is one in which any state, once reached, will with probability 1 be reached again. Aperiodicity is the condition that for each state  $i$ ,  $\gcd\{n : \text{state } i \text{ can be returned to in } n \text{ steps}\} = 1$ . Finally, an irreducible chain is one in which any state can somehow be reached from any other state.

<sup>3</sup>Up to multiplication by a constant. For the eigenvector  $\mathbf{v}$  to make sense as a probability distribution, it must be normalized so that its entries sum to 1.

converge after repeated applications of (3). For  $A$ , this eigenvector is

$$\mathbf{v} = \begin{bmatrix} \frac{\tau_{10}}{\tau_{01} + \tau_{10}} \\ \frac{\tau_{01}}{\tau_{01} + \tau_{10}} \end{bmatrix}. \quad (5)$$

This vector shows that the expected proportion of time the donkey spends in state 0 is  $\frac{\tau_{10}}{\tau_{01} + \tau_{10}}$ , and the expected proportion of time he spends in state 1 is  $\frac{\tau_{01}}{\tau_{01} + \tau_{10}}$ . These proportions motivate the conditional probability tree in Figure 1.



**Figure 1:** Conditional probability tree for donkey on a line

The first two branches of the tree show the marginal probabilities of the donkey being in either state, as given by the eigenvector  $\mathbf{v}$ . The second layer of branches identifies the conditional probabilities of the donkey staying in his current state or transitioning to the other. These probabilities come simply from the problem's construction. By the independence of the Markov chain, we can multiply connected branches to produce the final, joint probabilities of the tree. For example, the probability of the donkey being in state 0 and then switching to state 1 is  $\frac{\tau_{01}\tau_{10}}{\tau_{01} + \tau_{10}}$ . Notice this is equal to the probability of the donkey being in state 1 and then switching to state 0. Therefore, we define our frequency of transition as

$$\omega = \frac{\tau_{01}\tau_{10}}{\tau_{01} + \tau_{10}}. \quad (6)$$

We may think of  $\omega$  as the probability of observing some state transition at a given time step. This is our first statistic for use in a method of moments couplet.

### 2.1.2 Continuous Dynamics

Although the donkey's state is discrete, his position is continuous in both space and time. We let  $P(x, t)$  be the probability density of the donkey's position,  $x$ , at time  $t$ . Note that for fixed  $t$ ,  $P(x, t)$  is a probability density of one variable. Furthermore, since the donkey cannot simultaneously be in both states, we can express  $P$  as the sum of state-dependent, conditional probabilities,<sup>4</sup>  $P_0(x, t)$  and  $P_1(x, t)$ :

$$P(x, t) = P_0(x, t) + P_1(x, t). \quad (7)$$

Over time, the probability associated with state 0 is advected towards 0 and the probability associated with state 1 is advected towards 1. Probability is also transferred among states by the switching probabilities,  $\tau_{01}$  and  $\tau_{10}$ . These changes in probability density are captured by the conservation conditions

$$\frac{\partial P_0}{\partial t} = -\frac{\partial}{\partial x} [v(-x)P_0(x, t)] - \tau_{01}P_0(x, t) + \tau_{10}P_1(x, t) \quad (8)$$

$$\frac{\partial P_1}{\partial t} = -\frac{\partial}{\partial x} [v(1-x)P_1(x, t)] + \tau_{01}P_0(x, t) - \tau_{10}P_1(x, t). \quad (9)$$

We are interested in the long term behavior of the system, and thus we look for time-invariant steady-state solutions. That is, we examine the case in which

$$\frac{\partial P_0}{\partial t} = \frac{\partial P_1}{\partial t} = 0, \quad (10)$$

and we can, therefore, drop the dependence of  $P_0$  and  $P_1$  on  $t$ . Furthermore, we anticipate that in a steady-state solution, total fluxes are balanced:

$$v(-x)P_0(x) = v(1-x)P_1(x) \quad (11)$$

$$P_0(x) = \frac{1-x}{x}P_1(x). \quad (12)$$

The steady-state condition in state 1 is now

$$0 = \frac{\partial P_1}{\partial t} \quad (13)$$

$$= -\frac{\partial}{\partial x} [v(1-x)P_1(x)] + \tau_{01}\frac{1-x}{x}P_1(x) - \tau_{10}P_1(x). \quad (14)$$

---

<sup>4</sup>By referring to  $P_0$  and  $P_1$  as probabilities, we do not mean that their integrals over their support equal 1, but rather that they sum to a valid probability density function. If properly normalized,  $P_0$  and  $P_1$  would be the probability densities of the donkey's position given his being in state 0 or state 1, respectively.

Applying the derivative operator in (14) and rearranging terms, we have

$$-v(1-x) \frac{\partial P_1}{\partial x} + vP_1(x) = -\tau_{01} \frac{1-x}{x} P_1(x) + \tau_{10} P_1(x) \quad (15)$$

$$v(1-x) \frac{\partial P_1}{\partial x} = \tau_{01} \frac{1-x}{x} P_1(x) - \tau_{10} P_1(x) + vP_1(x) \quad (16)$$

$$\frac{\partial P_1}{\partial x} = \tau_{01} \frac{1}{vx} P_1(x) - \tau_{10} \frac{1}{v(1-x)} P_1(x) + \frac{1}{1-x} P_1(x). \quad (17)$$

This is a separable first-order ordinary differential equation. Using separation of variables we get

$$\frac{1}{P_1(x)} \partial P_1 = \left[ \frac{\tau_{01}}{v} \frac{1}{x} - \frac{\tau_{10}}{v} \frac{1}{1-x} + \frac{1}{(1-x)} \right] \partial x \quad (18)$$

$$\ln(P_1(x)) = \frac{\tau_{01}}{v} \ln(x) + \frac{\tau_{10}}{v} \ln(1-x) - \ln(1-x) + K \quad (19)$$

$$P_1(x) = Cx^{\frac{\tau_{01}}{v}} (1-x)^{\frac{\tau_{10}}{v}-1}. \quad (20)$$

Substituting  $P_1(x)$  into (12), we find that

$$P_0(x) = Cx^{\frac{\tau_{01}}{v}-1} (1-x)^{\frac{\tau_{10}}{v}}. \quad (21)$$

We then combine  $P_0$  and  $P_1$ , yielding the probability distribution

$$P(x) = P_0(x) + P_1(x) \quad (22)$$

$$= Cx^{\frac{\tau_{01}}{v}-1} (1-x)^{\frac{\tau_{10}}{v}-1}. \quad (23)$$

Thus, the constant

$$C = \frac{1}{\mathfrak{B}\left(\frac{\tau_{01}}{v}, \frac{\tau_{10}}{v}\right)}, \quad (24)$$

where

$$\mathfrak{B}\left(\frac{\tau_{01}}{v}, \frac{\tau_{10}}{v}\right) = \int_0^1 x^{\frac{\tau_{01}}{v}-1} (1-x)^{\frac{\tau_{10}}{v}-1} dx. \quad (25)$$

This makes  $P(x)$  a beta probability distribution (Johnson et al., 1995) with parameters  $\frac{\tau_{01}}{v}$  and  $\frac{\tau_{10}}{v}$ . The first two moments of  $P(x)$  are thus

$$\mu = \frac{\tau_{01}}{\tau_{01} + \tau_{10}} \quad (26)$$

and

$$\sigma^2 = \frac{\tau_{01}\tau_{10}}{(\tau_{01} + \tau_{10})^2 \left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)}. \quad (27)$$

The mean,  $\mu$ , and the variance,  $\sigma^2$ , provide two more statistics with which to form couplets.

### 2.1.3 Cumulative Power

A final statistic we examine is cumulative power. For a twice differentiable signal  $m$ , and finite  $t$ , cumulative power,  $F$ , is defined as

$$F(t) = \int_0^t (m''(s))^2 ds. \quad (28)$$

Cumulative power has been shown to be  $\mathcal{O}(t)$  for any  $m$  that is a finite sum of sine and cosine functions. Moreover, the average derivative of  $F$  for such an  $m$  is pivotal with respect to the frequency and the amplitude of  $m$  (Quinn, 2011). Thus, measurements of the rate of growth in  $F$  (with respect to time) can provide statistics amenable to method of moments type estimators.

In the donkey on a line system, we take  $m$  to be the donkey's position,  $x$ . Although  $x$  is not a periodic function, we derive a similar condition to the above for its cumulative power. First, however, we must adjust the definition (28) to account for the times  $t_1, \dots, t_n$  at which  $x$  is not twice differentiable. We note that  $t_1, \dots, t_n$  are the times at which the donkey changes state. From here on, we redefine the cumulative power of  $x$  as

$$F(t) = \int_0^{t_1} (x''(s))^2 ds + \int_{t_1}^{t_2} (x''(s))^2 ds + \dots + \int_{t_{n-1}}^{t_n} (x''(s))^2 ds + \int_{t_n}^t (x''(s))^2 ds, \quad (29)$$

where  $x'' = \frac{d^2x}{dt^2}$ . In Appendix D,  $F$  is shown to be  $\mathcal{O}(t)$ . Anticipating, then, that  $F(t)$  can be approximated by a line, we look to obtain another statistic for use in the method of moments approach: the expected value of  $\frac{dF}{dt}$ , in terms of  $\tau_{01}$  and  $\tau_{10}$ . In carrying out the calculation, we employ the gamma function  $\Gamma : \mathbb{R}^+ \rightarrow \mathbb{R}$  defined<sup>5</sup> by

$$\Gamma(a) = \int_0^\infty e^{-t} t^{a-1} dt. \quad (30)$$

Two useful properties of the gamma function (Abramowitz and Stegun, 1964) are

$$\Gamma(a) \Gamma(b) = \frac{\Gamma(a+b)}{a} \quad (31)$$

$$\Gamma(a+1) = a \Gamma(a). \quad (32)$$

From (32), we easily obtain the identity

$$\Gamma(a+2) = (a^2 + a) \Gamma(a). \quad (33)$$

Now beginning our derivation of the expected value of  $\frac{dF}{dt}$ , in state 0 we have

$$(x'')^2 = [(-vx)']^2 = (v^2x)^2 = v^4x^2, \quad (34)$$

---

<sup>5</sup> $\mathbb{R}^+$  denotes the set of positive real numbers. The gamma function can also be defined for complex numbers with positive real part (Abramowitz and Stegun, 1964).



and in state 1 we have

$$(x'')^2 = \left[ (v(1-x))' \right]^2 = (v^2(x-1))^2 = v^4(1-x)^2. \quad (35)$$

Let  $z$  be the donkey's current state. From (34) and (35), it is clear that  $\frac{dF}{dt}$  is conditional upon the donkey's current position and state. The expectation for  $\frac{dF}{dt}$ , therefore, is found by averaging  $\frac{dF}{dt}$  over all possible position and state combinations. The likelihood of observing any one of these combinations is conditional upon the parameters  $\tau_{01}$  and  $\tau_{10}$ . More formally,

$$\begin{aligned} E \left[ \frac{dF}{dt} \middle| \tau_{01}, \tau_{10} \right] &= \int_0^1 \left[ v^4 x^2 \Pr(x|z=0, \tau_{01}, \tau_{10}) \Pr(z=0|\tau_{01}, \tau_{10}) \right. \\ &\quad \left. + v^4 (1-x)^2 \Pr(x|z=1, \tau_{01}, \tau_{10}) \Pr(z=1|\tau_{01}, \tau_{10}) \right] dx. \end{aligned} \quad (36)$$

For ease of notation, we denote  $S = E \left[ \frac{dF}{dt} \middle| \tau_{01}, \tau_{10} \right]$ . Substituting known probabilities and probability densities, (36) becomes

$$\begin{aligned} S &= v^4 \left[ \int_0^1 x^2 \frac{1}{\beth\left(\frac{\tau_{01}}{v}, \frac{\tau_{10}}{v} + 1\right)} x^{\frac{\tau_{01}}{v}-1} (1-x)^{\frac{\tau_{10}}{v}} \left( \frac{\tau_{10}}{\tau_{01} + \tau_{10}} \right) dx \right. \\ &\quad \left. + \int_0^1 (1-x)^2 \frac{1}{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v}\right)} x^{\frac{\tau_{01}}{v}} (1-x)^{\frac{\tau_{10}}{v}-1} \left( \frac{\tau_{01}}{\tau_{01} + \tau_{10}} \right) dx \right]. \end{aligned} \quad (37)$$

We can multiply by fractions equal to 1:

$$\begin{aligned} S &= v^4 \left[ \int_0^1 \frac{1}{\beth\left(\frac{\tau_{01}}{v}, \frac{\tau_{10}}{v} + 1\right)} \frac{\beth\left(\frac{\tau_{01}}{v} + 2, \frac{\tau_{10}}{v} + 1\right)}{\beth\left(\frac{\tau_{01}}{v} + 2, \frac{\tau_{10}}{v} + 1\right)} x^{\frac{\tau_{01}}{v}+1} (1-x)^{\frac{\tau_{10}}{v}} \left( \frac{\tau_{01}}{\tau_{01} + \tau_{10}} \right) dx \right. \\ &\quad \left. + \int_0^1 \frac{1}{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v}\right)} \frac{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v} + 2\right)}{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v} + 2\right)} x^{\frac{\tau_{01}}{v}} (1-x)^{\frac{\tau_{10}}{v}+1} \left( \frac{\tau_{01}}{\tau_{01} + \tau_{10}} \right) dx \right]. \end{aligned} \quad (38)$$

Factoring out certain constants in (38) we get

$$\begin{aligned} S &= v^4 \left[ \frac{\beth\left(\frac{\tau_{01}}{v} + 2, \frac{\tau_{10}}{v} + 1\right)}{\beth\left(\frac{\tau_{01}}{v}, \frac{\tau_{10}}{v} + 1\right)} \left( \frac{\tau_{10}}{\tau_{01} + \tau_{10}} \right) \int_0^1 \frac{1}{\beth\left(\frac{\tau_{01}}{v} + 2, \frac{\tau_{10}}{v} + 1\right)} x^{\frac{\tau_{01}}{v}+1} (1-x)^{\frac{\tau_{10}}{v}} dx \right. \\ &\quad \left. + \frac{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v} + 2\right)}{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v}\right)} \left( \frac{\tau_{01}}{\tau_{01} + \tau_{10}} \right) \int_0^1 \frac{1}{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v} + 2\right)} x^{\frac{\tau_{01}}{v}} (1-x)^{\frac{\tau_{10}}{v}+1} dx \right]. \end{aligned} \quad (39)$$

Recognizing that the integrals in (39) are of densities taken over their support and are thus equal to 1, we are left with

$$S = v^4 \left[ \frac{\beth\left(\frac{\tau_{01}}{v} + 2, \frac{\tau_{10}}{v} + 1\right)}{\beth\left(\frac{\tau_{01}}{v}, \frac{\tau_{10}}{v} + 1\right)} \left( \frac{\tau_{10}}{\tau_{01} + \tau_{10}} \right) + \frac{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v} + 2\right)}{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v}\right)} \left( \frac{\tau_{01}}{\tau_{01} + \tau_{10}} \right) \right]. \quad (40)$$

Applying (31) to the first term of the right-hand side of (40) yields

$$\frac{\beth\left(\frac{\tau_{01}}{v} + 2, \frac{\tau_{10}}{v} + 1\right)}{\beth\left(\frac{\tau_{01}}{v}, \frac{\tau_{10}}{v} + 1\right)} \left(\frac{\tau_{10}}{\tau_{01} + \tau_{10}}\right) = \frac{\Gamma\left(\frac{\tau_{01}}{v} + 2\right) \Gamma\left(\frac{\tau_{10}}{v} + 1\right) \frac{1}{\Gamma\left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 3\right)}}{\Gamma\left(\frac{\tau_{01}}{v}\right) \Gamma\left(\frac{\tau_{10}}{v} + 1\right) \frac{1}{\Gamma\left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)}} \left(\frac{\tau_{10}}{\tau_{01} + \tau_{10}}\right) \quad (41)$$

$$= \frac{\Gamma\left(\frac{\tau_{01}}{v} + 2\right) \Gamma\left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)}{\Gamma\left(\frac{\tau_{01}}{v}\right) \Gamma\left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 3\right)} \left(\frac{\tau_{10}}{\tau_{01} + \tau_{10}}\right). \quad (42)$$

Now, by applying (33) to (42) we have

$$\frac{\beth\left(\frac{\tau_{01}}{v} + 2, \frac{\tau_{10}}{v} + 1\right)}{\beth\left(\frac{\tau_{01}}{v}, \frac{\tau_{10}}{v} + 1\right)} \left(\frac{\tau_{10}}{\tau_{01} + \tau_{10}}\right) = \frac{\left(\frac{\tau_{01}}{v}\right)^2 + \frac{\tau_{01}}{v}}{\left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)^2 + \left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)} \left(\frac{\tau_{10}}{\tau_{01} + \tau_{10}}\right). \quad (43)$$

A similar computation on the second term of the right-hand side of (40) yields

$$\frac{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v} + 2\right)}{\beth\left(\frac{\tau_{01}}{v} + 1, \frac{\tau_{10}}{v}\right)} \left(\frac{\tau_{01}}{\tau_{01} + \tau_{10}}\right) = \frac{\left(\frac{\tau_{10}}{v}\right)^2 + \frac{\tau_{10}}{v}}{\left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)^2 + \left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)} \left(\frac{\tau_{01}}{\tau_{01} + \tau_{10}}\right). \quad (44)$$

Now, substituting (43) and (44) into (40), we have

$$S = v^4 \left[ \frac{\left(\left(\frac{\tau_{01}}{v}\right)^2 + \frac{\tau_{01}}{v}\right) \tau_{10} + \left(\left(\frac{\tau_{10}}{v}\right)^2 + \frac{\tau_{10}}{v}\right) \tau_{01}}{(\tau_{01} + \tau_{10}) \left[\left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)^2 + \left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right)\right]} \right]. \quad (45)$$

Finally, simplifying (45) gives

$$S = v^4 \left[ \frac{\frac{\tau_{01}\tau_{10}}{v} \left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 2\right)}{(\tau_{01} + \tau_{10}) \left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1\right) \left(\frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 2\right)} \right] \quad (46)$$

$$= \frac{v^4 \tau_{01} \tau_{10}}{(\tau_{01} + \tau_{10})(\tau_{01} + \tau_{10} + v)}. \quad (47)$$

Therefore, the average slope of the cumulative power of  $x$  is

$$S = \frac{v^4 \tau_{01} \tau_{10}}{(\tau_{01} + \tau_{10})(\tau_{01} + \tau_{10} + v)}, \quad (48)$$

giving us our final statistic for use in a couplet.

### 2.1.4 Couplet Inversion and Results

Now we have derived closed-form expressions for mean, variance, frequency, and the average slope of cumulative power, in terms of  $\tau_{01}$  and  $\tau_{10}$ . The method of moments approach demands, however, that each of these statistics can be measured given the data. For mean and variance, this is clearly possible. Measuring frequency, though, requires accurate detection of state transitions. Finally, cumulative power is perhaps the most difficult to measure, as it requires knowledge of the donkey's state at all time steps. In the following simulations, we ignore these issues as to identify which couplets perform most effectively. That is, we assume we have perfect measurements of the four chosen statistics.

To complete our method of moments approach, we find invertible couplets of our closed-form expressions. Examining the equation for each statistic ((6), (26), (27), and (48)), we find that each one is symmetric with respect to  $\tau_{01}$  and  $\tau_{10}$ , with the exception of the mean,  $\mu$ . This implies that any invertible couplet will include the mean. For example, inverting the equations for mean,  $\mu = \frac{\tau_{01}}{\tau_{01} + \tau_{10}}$ , and frequency,  $\omega = \frac{\tau_{01} \tau_{10}}{\tau_{01} + \tau_{10}}$ , we find

$$\tau_{01} = \frac{\omega}{1 - \mu} \quad (49a)$$

$$\tau_{10} = \frac{\omega}{\mu}. \quad (49b)$$

Inverting the equations for mean and variance,  $\sigma^2 = \frac{\tau_{01} \tau_{10}}{(\tau_{01} + \tau_{10})^2 \left( \frac{\tau_{01}}{v} + \frac{\tau_{10}}{v} + 1 \right)}$ , we obtain

$$\tau_{01} = \frac{\mu^2 v (1 - \mu)}{\sigma^2} - \mu v \quad (50a)$$

$$\tau_{10} = \frac{v(\mu - 1)(\mu^2 - \mu + \sigma^2)}{\sigma^2}. \quad (50b)$$

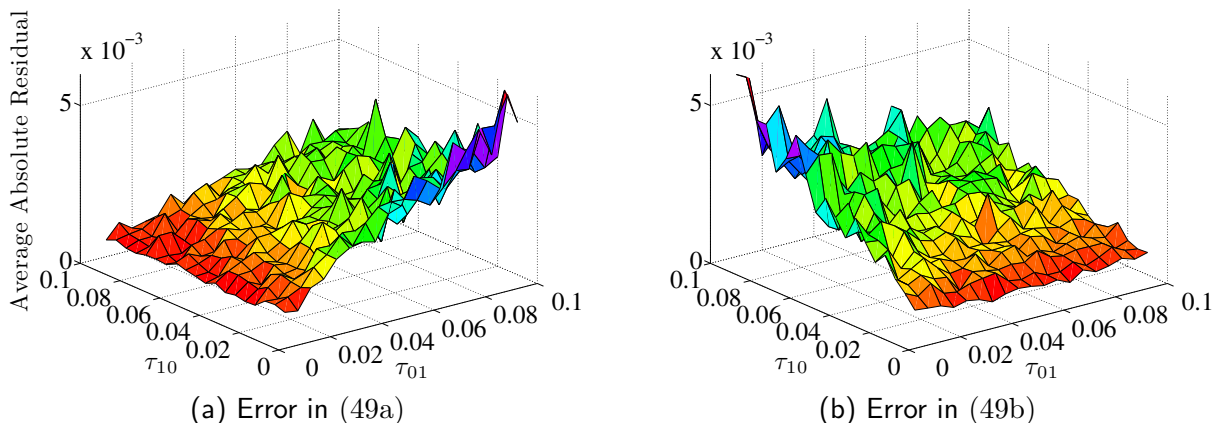
Finally, inverting the equations for mean and average cumulative power slope,  $S = \frac{v^4 \tau_{01} \tau_{10}}{(\tau_{01} + \tau_{10})(\tau_{01} + \tau_{10} + v)}$ , we find

$$\tau_{01} = \frac{\mu S v}{v^4 (\mu - \mu^2 - S)} \quad (51a)$$

$$\tau_{10} = \frac{S v (\mu - 1)}{\mu^2 v^4 - \mu v^4 + S}. \quad (51b)$$

Each of these couplet inversions vary in accuracy and precision, depending on the true values of  $\tau_{01}$  and  $\tau_{10}$ . For instance, Figure 2 shows how well the mean and frequency couplet (49) performs for different pairs of  $\tau_{01}$  and  $\tau_{10}$ . The surfaces shown were generated by running twenty simulations of donkey on a line, each one for 20,000 time steps, for every pair of parameters. There were 361 total pairs, created by meshing grids of 19 equally spaced points from 0.01 to 0.1. In each diagram, the vertical axis identifies the average

absolute error seen in the estimates (49a) and (49b). We can see that estimates improve as  $\tau_{01}$  and  $\tau_{10}$  both approach zero. Similar behavior was observed for the couplets (50) and (51).



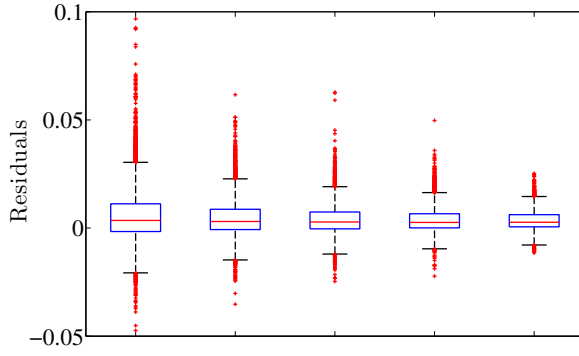
**Figure 2:** Average absolute residuals in parameter estimations using the mean and frequency couplet (49)

Next we compare the different couplets to one another. Figure 3 shows boxplots of the residuals for the estimators (49), (50), and (51). These boxplots were generated in the same manner as the results of Figure 2, except simulations were run with a varying number of total time steps. As the number of observed time steps increases, estimates become more accurate. This is due to the law of large numbers: As the donkey takes more steps, his observable statistics—mean, variance, frequency, and cumulative power, in our case—approach their theoretical values, improving the accuracy of the inverted couplets.

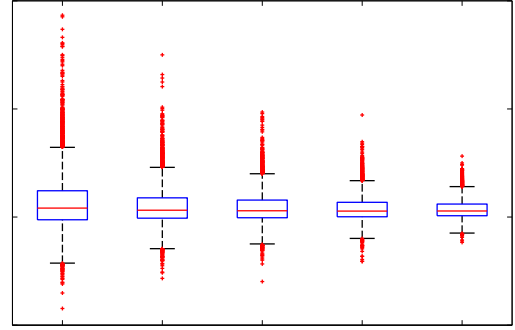
Based on these boxplots, the mean and frequency couplet produces, on average, the best estimates of  $\tau_{01}$  and  $\tau_{10}$ . The median residual (shown as the middle, red line) appears to converge to zero quickly, and the interquartile range (the vertical distance between the two blue lines) is narrow even for a small number of time steps. These qualities indicate that the mean and frequency couplet is a more efficient and less biased estimator of  $\tau_{01}$  and  $\tau_{10}$  than the other two couplets.

## 2.2 Likelihood-Based Estimation

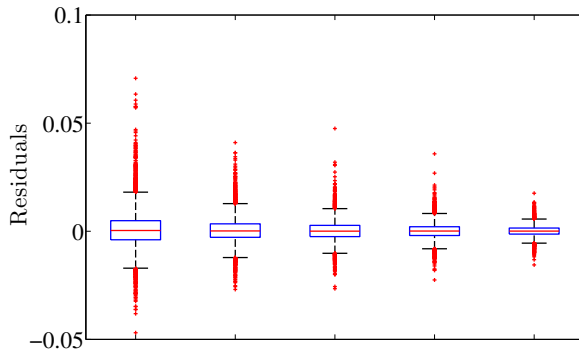
Another method of parameter estimation takes a likelihood-based approach. This approach involves calculating a function,  $F(\widehat{\tau}_{01}, \widehat{\tau}_{10})$ , that measures the likelihood of the observed data matching the parameters  $\widehat{\tau}_{01}$  and  $\widehat{\tau}_{10}$ . One such function is the log-likelihood function. Given a set of data, the log-likelihood function evaluates the beta probability density



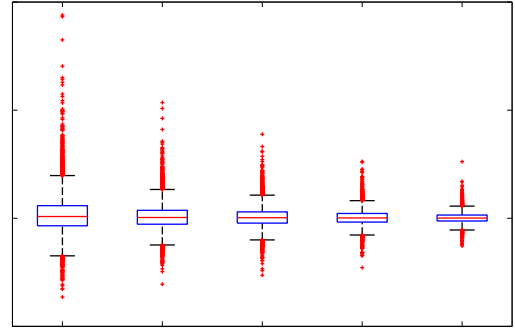
(a) Mean and Variance,  $\tau_{01}$



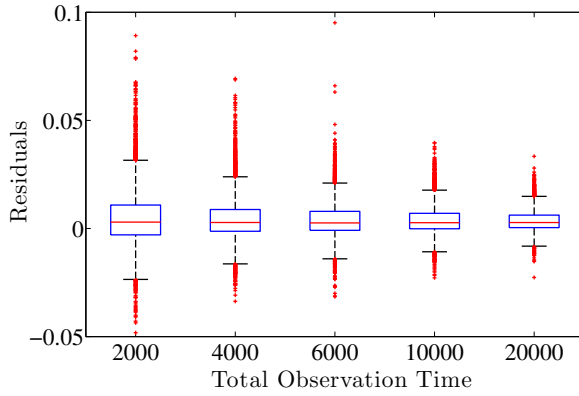
(b) Mean and Variance,  $\tau_{10}$



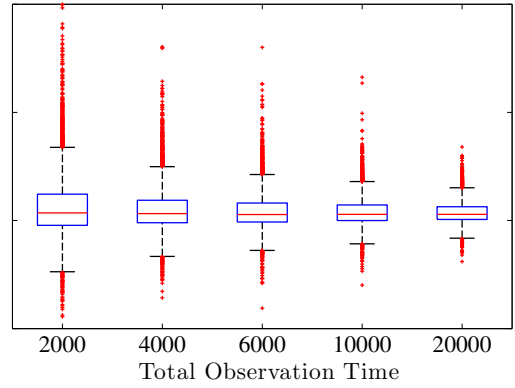
(c) Mean and Frequency,  $\tau_{01}$



(d) Mean and Frequency,  $\tau_{10}$



(e) Mean and Cumulative Power,  $\tau_{01}$



(f) Mean and Cumulative Power,  $\tau_{10}$

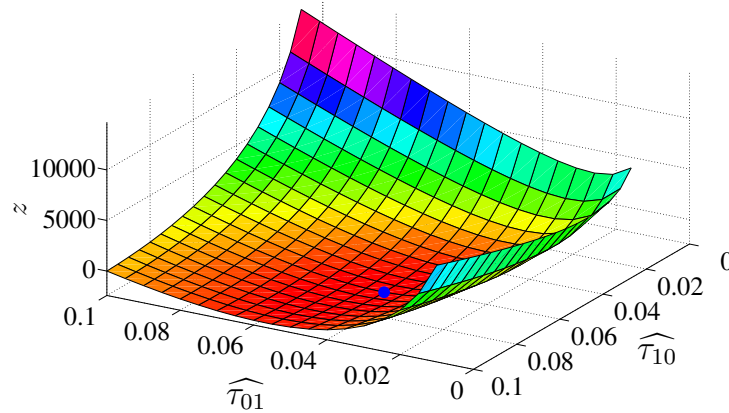
**Figure 3:** Boxplots comparing the performance of various couplets. Red markers denote outliers, defined as those values separated from the median by more than 1.5 times the interquartile range.

function

$$P(x|\widehat{\tau}_{01}, \widehat{\tau}_{10}) = Cx^{\frac{\widehat{\tau}_{01}}{v}-1}(1-x)^{\frac{\widehat{\tau}_{10}}{v}-1} \quad (52)$$

at the observed data (the donkey's observed positions). The function then takes the log of each result and sums these new results. Ideally, when this process is performed with the true  $\tau_{01}$  and  $\tau_{10}$ , the majority of data is concentrated in areas of high density, so that  $F(\tau_{01}, \tau_{10})$  is maximized at  $(\tau_{01}, \tau_{10})$ . To make this maximum a minimum, we examine  $-F(\tau_{01}, \tau_{10})$  instead of  $F(\tau_{01}, \tau_{10})$ . The estimates for  $\tau_{01}$  and  $\tau_{10}$  are taken to be the location of this minimum.

For example, Figure 4 graphs the negative log-likelihood function given a simulation of donkey on a line with 10,000 time steps. The input parameters were  $\tau_{01} = 0.05$  and  $\tau_{10} = 0.08$ , with  $v = 0.1$ . The minimum of the negative log-likelihood function, shown in Figure 4 as a blue dot, is located at  $\widehat{\tau}_{01} = 0.045$  and  $\widehat{\tau}_{10} = 0.085$ . Thus, estimation errors are quite low. And as with couplet inversion, this likelihood-based technique benefits from greater observation time.



**Figure 4:** Negative log-likelihood function with  $z = -F(\widehat{\tau}_{01}, \widehat{\tau}_{10})$

## 2.3 State Detection

While method of moments type estimators and likelihood-based estimators provide reasonable estimates for  $\tau_{01}$  and  $\tau_{10}$ , we are, in fact, able to calculate these parameters directly from our data. By comparing two consecutive positions of the donkey, we have a vector that points in the direction the donkey is moving. If the donkey's position is increasing, we know he is in state 1. By contrast, if the donkey's position is decreasing, we know he is in state 0. By dividing the number of time steps that the donkey changes from state 0 to state 1 by the number of time steps the donkey spends in state 0, we have our result for

$\tau_{01}$ . A similar computation yields  $\tau_{10}$ . This state detection technique provides the actual, observed values of  $\tau_{01}$  and  $\tau_{10}$ . Because of the stochastic nature of the system, we do not expect these measured values to exactly match our input values. Nevertheless, by the law of large numbers, a greater number of observed time steps should lend measured values closer to the input values. Table 1 affirms this expectation.

**Table 1:** Comparison of input and observed values for  $\tau_{ij}$

	Input ( $\times 10^{-3}$ )	Observed ( $\times 10^{-3}$ )	
		$N = 10000$	$N = 100000$
$\tau_{01}$	5.00	4.73	4.90
$\tau_{10}$	8.00	6.83	8.09

### 3 Donkey in a Triangle

We now consider a system in which the donkey has more than two states by extending the classic, one-dimensional problem to two dimensions. We let the donkey roam in a triangular pen with vertices located at the coordinates  $(0, 0)$ ,  $(1, 0)$ , and  $(0, 1)$ . Let state 0 be the state in which the donkey is moving toward  $(0, 0)$ , state 1 be the state in which the donkey is moving toward  $(1, 0)$ , and state 2 be the state in which the donkey is moving toward  $(0, 1)$ .<sup>6</sup> The donkey's continuous motion is modeled by the following equations:

$$\text{State 0: } \frac{dx}{dt} = v(-x) \quad \frac{dy}{dt} = v(-y) \quad (53)$$

$$\text{State 1: } \frac{dx}{dt} = v(1-x) \quad \frac{dy}{dt} = v(-y) \quad (54)$$

$$\text{State 2: } \frac{dx}{dt} = v(-x) \quad \frac{dy}{dt} = v(1-y). \quad (55)$$

Keeping the notation of the one-dimensional case, the donkey switches from state  $i$  to state  $j$  at a given time step with probability  $\tau_{ij}$ . We now have a Markov process with six parameters to estimate instead of only two.

#### 3.1 The Markov Process

Recall that in the one-dimensional case we examined a Markov chain to determine the expected proportion of time the donkey spends in each state. In the two-dimensional case we analyze an analogous process:

$$p_n = Ap_{n-1}, \quad (56)$$

---

<sup>6</sup>For simplicity, the donkey is always initially placed at  $(\frac{1}{3}, \frac{1}{3})$  and is taken to be in state 0.

where

$$A = \begin{bmatrix} 1 - \tau_{01} - \tau_{02} & \tau_{10} & \tau_{20} \\ \tau_{01} & 1 - \tau_{10} - \tau_{12} & \tau_{21} \\ \tau_{02} & \tau_{12} & 1 - \tau_{20} - \tau_{21} \end{bmatrix} \quad (57)$$

is still a column stochastic matrix. As before,  $A$  has eigenvalue 1 and associated eigenvector (see Appendix B)

$$\mathbf{v} = \begin{bmatrix} \frac{\tau_{21}\tau_{10} + \tau_{12}\tau_{20} + \tau_{10}\tau_{20}}{\tau_{21}\tau_{10} + \tau_{12}\tau_{20} + \tau_{10}\tau_{20} + \tau_{20}\tau_{01} + \tau_{02}\tau_{21} + \tau_{01}\tau_{21} + \tau_{10}\tau_{02} + \tau_{01}\tau_{12} + \tau_{02}\tau_{12}} \\ \frac{\tau_{20}\tau_{01} + \tau_{02}\tau_{21} + \tau_{01}\tau_{21}}{\tau_{21}\tau_{10} + \tau_{12}\tau_{20} + \tau_{10}\tau_{20} + \tau_{20}\tau_{01} + \tau_{02}\tau_{21} + \tau_{01}\tau_{21} + \tau_{10}\tau_{02} + \tau_{01}\tau_{12} + \tau_{02}\tau_{12}} \\ \frac{\tau_{10}\tau_{02} + \tau_{01}\tau_{12} + \tau_{02}\tau_{12}}{\tau_{21}\tau_{10} + \tau_{12}\tau_{20} + \tau_{10}\tau_{20} + \tau_{20}\tau_{01} + \tau_{02}\tau_{21} + \tau_{01}\tau_{21} + \tau_{10}\tau_{02} + \tau_{01}\tau_{12} + \tau_{02}\tau_{12}} \end{bmatrix}. \quad (58)$$

In the one-dimensional case, we were able to use the eigenvector associated with the Markov matrix to determine the frequency of the donkey switching between states. By adding just one more state to our system, it becomes difficult to define what is meant by the term “frequency.” We now have several types of transitions, so using frequency as an intuitive, feature statistic is less appealing. Furthermore, as we continue to expand the system, finding a closed form for the eigenvector becomes computationally unrealistic; with just six states, the number of terms in the numerator of an eigenvector coordinate exceeds 1000 (see Appendix C).

### 3.2 Continuous Dynamics

Hoping to derive statistics such as mean and variance in the two-dimensional case, we are interested in finding the probability distribution of our system. We let  $P(x, y, t)$  be the probability density of the donkey’s position at time  $t$ . Again, we consider a decomposition of  $P$  into the state-dependent, conditional probabilities  $P_0(x, y, t)$ ,  $P_1(x, y, t)$ , and  $P_2(x, y, t)$ :

$$P(x, y, t) = P_0(x, y, t) + P_1(x, y, t) + P_2(x, y, t). \quad (59)$$

We have the following conservation conditions, which are direct extensions of (8) and (9):

$$\frac{\partial P_0}{\partial t} = -\frac{\partial}{\partial x} [v(-x)P_0] - \frac{\partial}{\partial y} [v(-y)P_0] - \tau_{01}P_0 - \tau_{02}P_0 + \tau_{10}P_1 + \tau_{20}P_2 \quad (60)$$

$$\frac{\partial P_1}{\partial t} = -\frac{\partial}{\partial x} [v(1-x)P_1] - \frac{\partial}{\partial y} [v(-y)P_1] - \tau_{10}P_1 - \tau_{12}P_1 + \tau_{01}P_0 + \tau_{21}P_2 \quad (61)$$

$$\frac{\partial P_2}{\partial t} = -\frac{\partial}{\partial x} [v(-x)P_2] - \frac{\partial}{\partial y} [v(1-y)P_2] - \tau_{20}P_2 - \tau_{21}P_2 + \tau_{02}P_0 + \tau_{12}P_1. \quad (62)$$

As in the one-dimensional case, we look for a steady-state solution, when

$$\frac{\partial P_0}{\partial t} = \frac{\partial P_1}{\partial t} = \frac{\partial P_2}{\partial t} = 0. \quad (63)$$



That is, the probability distribution of the location of the donkey does not change with time, and we may drop the dependence of  $P_0$ ,  $P_1$ , and  $P_2$  on  $t$ . Adding (60), (61), and (62) in this case, we have

$$0 = -\frac{\partial}{\partial x} [v(-x)P_0] - \frac{\partial}{\partial x} [v(1-x)P_1] - \frac{\partial}{\partial x} [v(-x)P_2] \\ - \frac{\partial}{\partial y} [v(-y)P_0] - \frac{\partial}{\partial y} [v(-y)P_1] - \frac{\partial}{\partial y} [v(1-y)P_2], \quad (64)$$

where we have collected the  $x$  and  $y$  components. Assuming total fluxes are balanced in each direction, (64) implies

$$0 = v(x)P_0 - v(1-x)P_1 + v(x)P_2 \quad (65)$$

$$0 = v(y)P_0 + v(y)P_1 - v(1-y)P_2. \quad (66)$$

Solving for  $P_0$  in (65) yields

$$P_0 = -P_1 + \frac{P_1}{x} - P_2. \quad (67)$$

Substituting (67) into (66) gives us

$$0 = v(y) \left( -P_1 + \frac{P_1}{x} - P_2 \right) + v(y)P_1 - v(1-y)P_2 \quad (68)$$

$$P_1 = \frac{xP_2}{y}. \quad (69)$$

Using (69) in (67), we see

$$P_0 = -\frac{xP_2}{y} + \frac{P_2}{y} - P_2. \quad (70)$$

We then apply (69) and (70) to (62), the left-hand side of which we have assumed to be 0:

$$0 = -\frac{\partial}{\partial x} [v(-x)P_2] - \frac{\partial}{\partial y} [v(1-y)P_2] - \tau_{20}P_2 - \tau_{21}P_2 \\ + \tau_{02} \left( -\frac{xP_2}{y} + \frac{P_2}{y} - P_2 \right) + \tau_{12} \frac{xP_2}{y}. \quad (71)$$

Applying the derivative operators using the product rule and rearranging terms, we arrive at the following differential equation:

$$-x \frac{\partial P_2}{\partial x} + (1-y) \frac{\partial P_2}{\partial y} = \frac{P_2}{v} \left( -\tau_{20} - \tau_{21} + \tau_{02} + \frac{\tau_{02}}{y} + \frac{(\tau_{12} - \tau_{02})x}{y} + 2v \right). \quad (72)$$

Although we identify (72) as a first-order, semilinear partial differential equation (Polyanin and Zaitsev, 2004), we are currently not able to solve it for  $P_2$ . In fact, we are not sure this is a well-posed problem. Unfortunately, this means we cannot apply the same methods as before to find expressions for moments such as mean and variance. Moreover, we have no probability distribution with which to perform the same likelihood-based scheme as described in 2.2.

### 3.3 State Detection

As some of the methods applied to the donkey on a line system become intractable in a multi-dimensional extension, we turn to state detection as in 2.3. Before applying a method of state detection to the donkey in a triangular pen, however, we check that his movements are always linear. Consider the case when the donkey is in state 0. As stated earlier, his movement in the  $x$  direction is given by  $\frac{dx}{dt} = v(-x)$ , and his movement in the  $y$  direction is given by  $\frac{dy}{dt} = v(-y)$ . Dividing these, we find

$$\frac{\frac{dy}{dt}}{\frac{dx}{dt}} = \frac{dy}{dx} = \frac{v(-y)}{v(-x)} = \frac{y}{x}. \quad (73)$$

Since the slope given by (73) is exactly in the direction of  $(0, 0)$ , we see that the donkey's movement is linear. A similar computation can be performed for states 1 and 2 to show his movement is linear in every state. From this fact, we build a state detector for the donkey using the following algorithm:

1. Compare two consecutive positions of the donkey to make a vector pointing in the direction of the donkey's current movement; call it  $\mathbf{a}$ .
2. Construct vectors pointing from the donkey's current position to each of the coordinates of the triangle; call them  $\mathbf{b}_i$  for  $i = 0, 1, 2$ .
3. Compare  $\mathbf{a}$  to each  $\mathbf{b}_i$  by calculating the cosine of the angle between them via the formula

$$\cos \theta = \frac{\mathbf{a} \cdot \mathbf{b}_i}{\|\mathbf{a}\| \|\mathbf{b}_i\|}. \quad (74)$$

The  $\mathbf{b}_i$  that yields  $\cos \theta = 1$  is the vector that is parallel to the donkey's movement. Thus, the donkey is in state  $i$ .

4. Repeat this process for all observed data.

Now that we have a state detector, we can use it to find each  $\tau_{ij}$  empirically. For example, if we want to estimate  $\tau_{10}$ , we first use the state detector to find the number of

time steps that the donkey spends in state 1. Next, find each time step that the donkey changes from state 1 to state 0. By dividing these results, we have our estimate for  $\tau_{10}$ . This process is easily seen to generalize to any convex geometry in which the donkey can roam.

Because of the stochastic nature of our system, we do not expect our observed  $\tau_{ij}$ 's to match perfectly with our input  $\tau_{ij}$ 's. However, by the law of large numbers, we expect that they will converge as the number of time steps increases. Table 2 shows the results from a simulation with  $v = 0.01$  and varied total observed time steps  $N$ . As expected, a greater number of observed time steps yields, on average, observed parameter values closer to input parameter values.

**Table 2:** Comparison of input and observed values for  $\tau_{ij}$

	Input ( $\times 10^{-3}$ )	Observed ( $\times 10^{-3}$ )	
		$N = 10000$	$N = 100000$
$\tau_{01}$	1.00	0.76	0.81
$\tau_{02}$	6.00	5.83	6.56
$\tau_{10}$	2.00	3.52	1.71
$\tau_{12}$	3.00	2.24	3.15
$\tau_{20}$	4.00	5.13	3.95
$\tau_{21}$	5.00	5.13	5.02

## 4 Noisy Data

The state detector solves our two-dimensional extension of Buridan's ass, as it allows us to directly measure the parameters of the system. Although these measured values may not be equal to the input parameters, they are the values we wish to obtain as we continue our exploration of donkey in a triangle. They are the observed, as opposed to theoretical, probabilities of transitioning, making them the target estimates for further simulations.

A system in which we have perfect measurements of the donkey's trajectory is not realistic. Real-world observations of dynamical systems involve some level of error, or noise. By adding noise to our system, we obtain data that is comparable to real-world measurements of dynamical systems. We now shift our focus to managing noise in our data, with the goal of obtaining reasonable estimates for the  $\tau_{ij}$ 's.

To compare fairly the performance of the techniques discussed, we apply each to the same data set. This data was taken from the simulation used to construct Table 2. From here on we will assume we observe the donkey for  $N = 10000$  time steps, so that the target estimates for the various  $\tau_{ij}$  are those shown in the middle column of Table 2. Independent and identically distributed noise was added to each coordinate of the donkey's trajectory,

at each time step. This noise was taken from a normal distribution with a mean of 0 and a standard derivation of 0.01.

## 4.1 Linear Regression

Linear regression is a statistical technique that aids in the study of linear relationships between variables. By modeling a set of data by a line, linear regression has potential to remove artifacts of noise. Outlined below is our application of linear regression to state detection:

1. Consider the donkey's observed positions in the  $x$  and  $y$  directions separately. We let  $\mathbf{x}$  and  $\mathbf{y}$  be vectors containing the positions of the donkey in these directions, so that  $(x_j, y_j)$  is the donkey's observed location at the  $j$ th time step. We let  $\mathbf{t}$  be the vector containing the times at which observations are made. In general, we assume  $t_j = j - 1$ .
2. For each  $j = 1, \dots, N+1$ , where  $N$  is the number of observed time steps, we calculate the lines of best fit to the data sets  $(t_k, x_k)_{k=j}^{j+W}$  and  $(t_k, y_k)_{k=j}^{j+W}$ , where  $W$  is the so-called viewing window. That is,  $W$  is the number of time steps from the current position considered in linear regression. (If  $j + W$  exceeds the length of  $\mathbf{x}$  or  $\mathbf{y}$ , then  $W$  is taken to be the largest possible viewing window.)
3. We let  $a_1$  be the slope of the best-fit line in the  $x$  direction, and  $a_2$  be the slope of the best-fit line in the  $y$  direction. We set  $\mathbf{a} = [a_1, a_2]^T$ .
4. Next construct vectors pointing from  $(\bar{x}, \bar{y})$ , where  $\bar{x} = \frac{1}{W+1} \sum_{k=j}^{j+W} x_k$ , and  $\bar{y} = \frac{1}{W+1} \sum_{k=j}^{j+W} y_k$ , to each of the coordinates of the triangle; call these vectors  $\mathbf{b}_i$  for  $i = 0, 1, 2$ .
5. Proceed as in 3.3, now accounting for noise by taking the  $i$  that yields the greatest value in (74).

Parameter estimates obtained using this algorithm are displayed in Table 3. Although increasing the viewing window  $W$  leads to more reasonable estimates, in no case does linear regression provide reliable information. Each estimate's absolute relative error is also shown in blue in Table 3, where the error is computed based on the observed  $\tau_{ij}$  values of Table 2. Clearly the noise present in the system must be dealt with in a different manner.

## 4.2 Denoising Techniques

An alternative approach to linear regression is to denoise the observed data before applying state detection. That is, through a chosen method, we transform our observed, noisy

**Table 3:** Estimated values ( $\times 10^{-3}$ ) for  $\tau_{ij}$  and **relative error** after adding noise, using linear regression with different viewing windows

	$W = 1$	$W = 3$	$W = 5$
$\tau_{01}$	415.67 (54471.04%)	248.34 (32581.83%)	160.30 (20995.21%)
$\tau_{02}$	416.29 (7045.70%)	239.39 (4009.18%)	131.80 (2162.38%)
$\tau_{10}$	372.51 (10492.96%)	214.55 (6001.10%)	138.62 (3841.92%)
$\tau_{12}$	412.57 (18336.12%)	193.85 (8562.11%)	100.55 (4392.25%)
$\tau_{20}$	372.17 (7154.76%)	206.50 (3925.43%)	122.19 (2281.89%)
$\tau_{21}$	402.30 (7742.08%)	191.39 (3630.75%)	94.52 (1742.60%)
	$W = 10$	$W = 15$	$W = 20$
$\tau_{01}$	54.47 (7068.42%)	28.32 (3626.27%)	22.17 (2817.15%)
$\tau_{02}$	53.31 (815.01%)	29.16 (400.47%)	22.17 (280.50%)
$\tau_{10}$	55.13 (1467.63%)	36.78 (945.85%)	29.14 (728.60%)
$\tau_{12}$	40.34 (1702.84%)	25.68 (1047.60%)	17.93 (701.28%)
$\tau_{20}$	57.25 (1015.92%)	26.85 (423.31%)	20.81 (305.74%)
$\tau_{21}$	37.06 (622.41%)	29.29 (470.88%)	20.51 (299.77%)

data into a data set which should represent the actual trajectory of the donkey. We now explore various methods of denoising and compare the accuracy of the resulting parameter estimates. Each method is presented as applied to a one-dimensional data set. That is, we denoise the data vectors  $\mathbf{x}$  and  $\mathbf{y}$  (each one containing  $N + 1$  coordinates) separately and then apply the state detector presented in 3.3 to this denoised data, thereby producing estimates of the various  $\tau_{ij}$ .

#### 4.2.1 Locally Weighted Polynomial Regression

The first denoising technique we explore is locally weighted polynomial regression (LWPR). LWPR is a method of fitting a curve to a data set using low-degree polynomials (Fan and Gijbels, 1996). To utilize the versatility of polynomials while avoiding the Runge phenomenon that plagues high-degree interpolation (Runge, 1901), LWPR fits polynomials to subsets of the data rather than fitting a single polynomial to the entire set. A smoothing process is then applied to join the local polynomials into a single curve. This process is governed by a smoothing parameter,  $0 < h \leq 1$ , which is the percentage of the total data that is considered in smoothing about a given data point (Cleveland, 1979).

In the same simulation as before, LWPR was performed using quadratic polynomials and  $h = 0.005$ . Results are shown in Table 4. Despite being far more reasonable than the estimates provided by linear regression, those provided by LWPR are still poor. The estimates are nearly correct in the ordering of the parameters (i.e. which are the smallest, and which are the largest), but they are, in every case, overestimates. Such distortion of

the parameters is caused by over-detection of state transitions. The performance of LWPR may be improved by utilizing polynomials of a different degree and adjusting the parameter  $h$ .

**Table 4:** Estimated values for  $\tau_{ij}$  and [relative error](#) after adding noise, using locally weighted polynomial regression

	Input ( $\times 10^{-3}$ )	Observed ( $\times 10^{-3}$ )	Estimate ( $\times 10^{-3}$ )
$\tau_{01}$	1.00	0.76	3.76 (395.40%)
$\tau_{02}$	6.00	5.83	9.41 (61.54%)
$\tau_{10}$	2.00	3.52	8.48 (141.06%)
$\tau_{12}$	3.00	2.24	4.89 (118.55%)
$\tau_{20}$	4.00	5.13	7.16 (39.50%)
$\tau_{21}$	5.00	5.13	8.40 (63.76%)

#### 4.2.2 Wavelet Filtering

Wavelets are functions that together form an orthonormal basis of an infinite-dimensional vector space<sup>7</sup> consisting of other functions. There are many different wavelet families, each suitable to represent a unique class of functions. While providing the convenience of orthonormal bases, wavelets have the added utility of decomposing functions by both frequency and scale (Mallat, 2008). That is, frequency can be analyzed over subsets of a function’s domain, in contrast to decomposition into a Fourier basis, where a function’s entire domain is considered in each basis coefficient.

Wavelets can be used to filter noisy signals by decomposing them into their orthonormal basis representation and adjusting coefficients polluted by noise. In our case, we use the symlets 8 wavelet basis (up to level 6) with multi-level, soft thresholding. Table 5 shows the parameter estimations resulting from this denoising. The overestimates suggest that wavelet filtering still allows over-detection of state transitions. There are many other wavelet families, however, and a variety of thresholding techniques. Wavelet filtering cannot be discarded entirely before performing a thorough exploration of the many possible combinations.

#### 4.2.3 Butterworth Filtering

The Butterworth filter is a signal processing tool used to denoise a system by permitting and dampening certain specified frequencies present in the data (Butterworth, 1930). There are three types of Butterworth filters: lowpass, highpass, and bandpass. A lowpass filter permits frequencies below a specified cutoff frequency  $\omega_c$  but gradually dampens those

---

<sup>7</sup>Often this vector space is  $L^2(\mathbb{R})$ , the space of square-integrable functions on the real line.

**Table 5:** Estimated values for  $\tau_{ij}$  and **relative error** after adding noise, using wavelet filtering

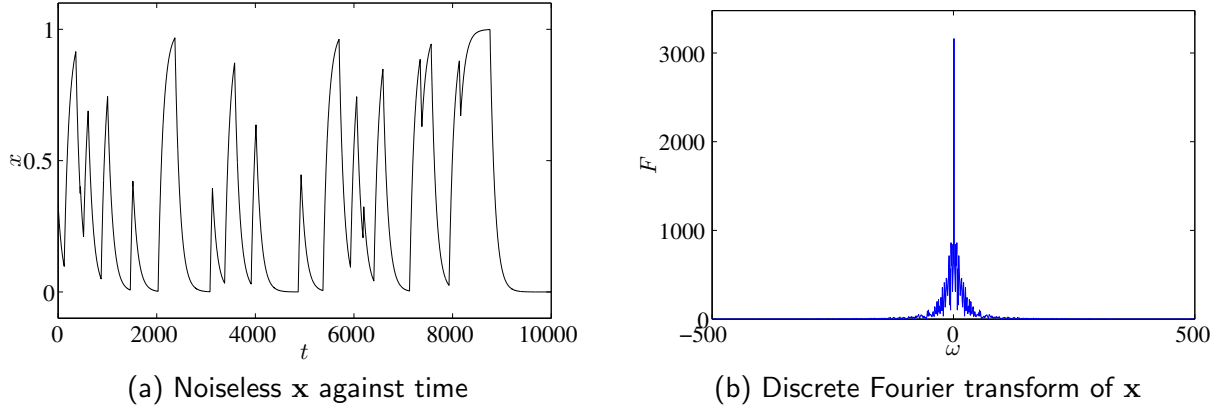
	Input ( $\times 10^{-3}$ )	Observed ( $\times 10^{-3}$ )	Estimate ( $\times 10^{-3}$ )
$\tau_{01}$	1.00	0.76	22.52 (2863.47%)
$\tau_{02}$	6.00	5.83	49.21 (744.66%)
$\tau_{10}$	2.00	3.52	28.40 (707.54%)
$\tau_{12}$	3.00	2.24	21.06 (841.05%)
$\tau_{20}$	4.00	5.13	51.39 (901.80%)
$\tau_{21}$	5.00	5.13	22.64 (341.27%)

exceeding  $\omega_c$ . In contrast, a highpass filter permits frequencies above the cutoff but dampens those below it. A bandpass filter has both lowpass and highpass cutoff frequencies.

The type of filter we apply depends on which frequencies are most prevalent in the observed data. Figure 5 shows that, at least in our sample simulation, such frequencies are low in the spectrum. The  $x$  coordinate of the donkey's trajectory is shown in Figure 5a, and the discrete Fourier transform  $F$  of  $\mathbf{x}$ , defined by

$$F(\omega) = \sum_{k=1}^{N+1} x_k e^{\frac{-2\pi\sqrt{-1}\omega(k-1)}{N+1}}, \quad (75)$$

is shown in Figure 5b. As the majority of energy is contained in the lower frequencies, a lowpass Butterworth filter is suitable for our problem. Our goal is to eliminate noise that may dominate high frequencies while relying on the high energy in the low frequencies to suppress the effect of noise there.



**Figure 5:** Identification of predominant frequencies in the donkey's trajectory

To denoise the donkey's trajectory, a fifth order lowpass Butterworth filter was used, with  $\omega_c = 100$ . The resulting parameter estimates are given in Table 6. These results are easily the best seen thus far.

**Table 6:** Estimated values for  $\tau_{ij}$  and [relative error](#) after adding noise, using butterworth filtering

	Input ( $\times 10^{-3}$ )	Observed ( $\times 10^{-3}$ )	Estimate ( $\times 10^{-3}$ )
$\tau_{01}$	1.00	0.76	2.68 (253.19%)
$\tau_{02}$	6.00	5.83	4.29 (26.29%)
$\tau_{10}$	2.00	3.52	4.14 (17.77%)
$\tau_{12}$	3.00	2.24	2.55 (13.89%)
$\tau_{20}$	4.00	5.13	4.57 (12.95%)
$\tau_{21}$	5.00	5.13	3.51 (31.60%)

#### 4.2.4 Total Variation Denoising

The last denoising approach we considered is a technique commonly used in image processing: total variation (TV) (Rudin et al., 1992). Unlike the Butterworth filter, TV is not spectral-based; that is, it does not involve decomposition into a Fourier basis. Rather, to denoise a vector  $\mathbf{x}$ , TV seeks to solve the optimization problem

$$\arg \min_{\hat{\mathbf{x}}} \frac{\gamma}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 + \|\nabla \hat{\mathbf{x}}\|_1, \quad (76)$$

where

$$\nabla = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 1 & 0 & \cdots & 0 & 0 \\ 0 & -1 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & -1 & 1 \end{bmatrix} \quad (77)$$

is a discrete derivative operator.<sup>8</sup> The goal is to capture the overall behavior of  $\mathbf{x}$  with an estimate  $\hat{\mathbf{x}}$ , while removing small, spurious oscillations that are merely artifacts of noise. Hence, we minimize the sum of a fitting term and a penalty term that discretely approximates the integral of the absolute value of the derivative (this integral is called a function’s *total variation*). Our choice of  $\gamma$  should reflect the relative importance of minimizing each term. With greater noise, oscillations become more pronounced, so  $\gamma$  would be placed at a lower value to assign more weight to minimizing total variation.

Difficulty lies in solving (76), due to the non-differentiability of the  $\ell_1$ -norm.<sup>9</sup> To resolve this issue, we refer to Goldstein and Osher (2009), in which the authors rephrase the

<sup>8</sup> $\nabla$  is shown with entries of  $\pm 1$  since we assume measurements are taken a unit of time apart. The matrix, however, can be generalized for nonunit and nonuniform spacing, each row with entries of  $\pm \Delta t$  for a unique  $\Delta t$ .

<sup>9</sup>The  $\ell_1$ -norm is defined as  $\|\mathbf{x}\|_1 = \sum_k |x_k|$ . The  $\ell_2$ -norm is defined as  $\|\mathbf{x}\|_2 = \sqrt{\sum_k |x_k|^2}$ .



unconstrained minimization into a constrained one, in the process splitting the  $\ell_1$  and  $\ell_2$  portions of the problem. In this new formulation, the authors solve the minimization using so-called Split Bregman iteration, which requires the convexity of the  $\ell_1$ - and  $\ell_2$ -norms. We demonstrate below this process as applied to (76).

We first reformulate (76) as an equivalent unconstrained problem:

$$\arg \min_{\hat{\mathbf{x}}} \frac{\gamma}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 + \|\mathbf{d}\|_1 \text{ such that } \mathbf{d} = \nabla \hat{\mathbf{x}}. \quad (78)$$

The constraint in (78), though, is weakly enforced, introducing a second fitting term with weight parameter  $\lambda$ :

$$\arg \min_{\hat{\mathbf{x}}} \frac{\gamma}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{d} - \nabla \hat{\mathbf{x}}\|_2^2 + \|\mathbf{d}\|_1. \quad (79)$$

The optimization as presented in (79) is now amenable to Split Bregman iteration, which initializes  $\mathbf{d}_0 = \mathbf{b}_0 = \mathbf{0}$  and follows a two step process for  $i = 1, \dots, n$ :

$$\text{Step 1: } (\hat{\mathbf{x}}_{i+1}, \mathbf{d}_{i+1}) = \arg \min_{\hat{\mathbf{x}}, \mathbf{d}} \frac{\gamma}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{d} - \nabla \hat{\mathbf{x}} - \mathbf{b}_i\|_2^2 + \|\mathbf{d}\|_1 \quad (80a)$$

$$\text{Step 2: } \mathbf{b}_{i+1} = \mathbf{b}_i + \nabla \hat{\mathbf{x}}_{i+1} - \mathbf{d}_{i+1}. \quad (80b)$$

For step 1, the algorithm solves for  $\hat{\mathbf{x}}$  and  $\mathbf{d}$  separately. First,

$$\hat{\mathbf{x}}_{i+1} = \arg \min_{\hat{\mathbf{x}}} \frac{\gamma}{2} \|\hat{\mathbf{x}} - \mathbf{x}\|_2^2 + \frac{\lambda}{2} \|\mathbf{d} - \nabla \hat{\mathbf{x}} - \mathbf{b}_i\|_2^2, \quad (81)$$

which can be solved by differentiating with respect to  $\hat{\mathbf{x}}$ , and setting the result equal to  $\mathbf{0}$ , as the right-hand side is convex. That is, the minimizer  $\hat{\mathbf{x}}$  is such that

$$\gamma (\hat{\mathbf{x}} - \mathbf{x})^T - \lambda (\mathbf{d}_i - \nabla \hat{\mathbf{x}} - \mathbf{b}_i)^T \nabla = \mathbf{0}^T, \quad (82)$$

or equivalently

$$(\gamma I + \nabla^T \nabla) \hat{\mathbf{x}} = \lambda \nabla^T (\mathbf{d}_i - \mathbf{b}_i) + \gamma \mathbf{x}. \quad (83)$$

We now attain the solution for (81) by solving for  $\hat{\mathbf{x}}$ :

$$\hat{\mathbf{x}}_i = (\gamma I + \nabla^T \nabla)^{-1} (\lambda \nabla^T (\mathbf{d}_i - \mathbf{b}_i) + \gamma \mathbf{x}). \quad (84)$$

Next, the algorithm produces  $\mathbf{d}_{i+1}$  coordinatewise by

$$d_{i+1,j} = \text{shrink}((\nabla \mathbf{c}_{i+1})_j + b_{i,j}, 1/\lambda), \quad (85)$$

where

$$\text{shrink}(x, \delta) = \frac{x}{|x|} \cdot \max(|x| - \delta, 0). \quad (86)$$

Once the  $n$  iterations of (80) are carried out, we have our denoised  $\hat{\mathbf{x}}$  in the form of (84).

The above algorithm was applied to our system with  $\gamma = 0.5$ ,  $\lambda = 20$ , and  $n = 10$ . The parameter estimates obtained are displayed in Table 7. These results are comparable to those resulting from Butterworth filtering. The parameter with the lowest input and lowest observed value,  $\tau_{01}$ , is still estimated with the greatest error. This error suggests parameters with smaller values come with greater uncertainty. To explain this, we note that a smaller parameter value implies the associated transition is observed less frequently, perhaps making estimation schemes more sensitive to noise.

**Table 7:** Estimated values for  $\tau_{ij}$  and **relative error** after adding noise, using total variation

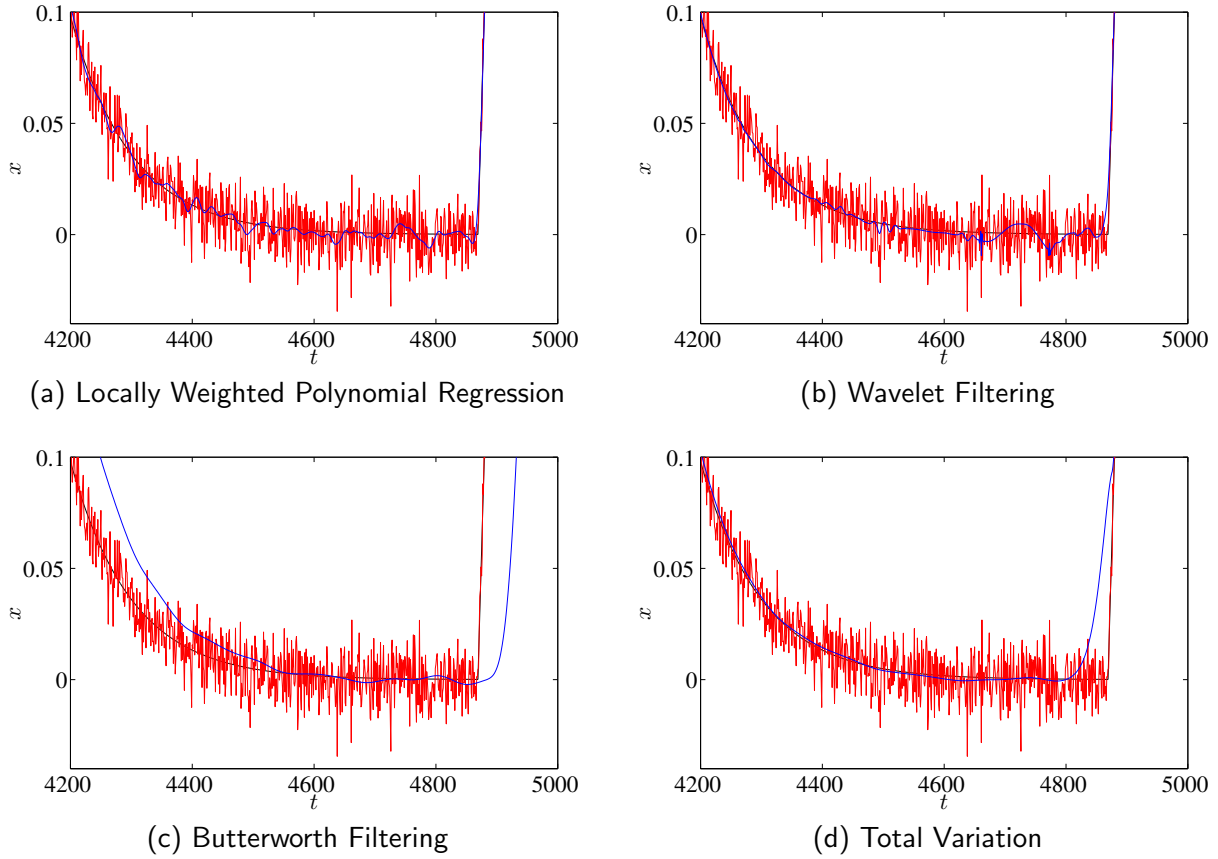
	Input ( $\times 10^{-3}$ )	Observed ( $\times 10^{-3}$ )	Estimate ( $\times 10^{-3}$ )
$\tau_{01}$	1.00	0.76	3.45 (374.69%)
$\tau_{02}$	6.00	5.83	4.44 (23.79%)
$\tau_{10}$	2.00	3.52	4.44 (26.30%)
$\tau_{12}$	3.00	2.24	2.86 (27.59%)
$\tau_{20}$	4.00	5.13	4.62 (9.86%)
$\tau_{21}$	5.00	5.13	3.08 (39.91%)

As can be seen in Table 7, the total variation approach tends to create parameter estimates exhibiting less variance than the actual, observed values. That is, parameters with high values are often underestimated, and parameters with low values are often overestimated. This may be a result of a marked shortcoming of TV: the ability to preserve sharp changes in the data. While TV eliminates the spurious oscillations created by noise, it follows state transitions too early. Figure 6d illustrates this feature.

Figure 6 also provides a graphical explanation for the deficiency of LWPR and wavelet filtering. The artificial oscillations created by noise are not fully removed (seen in Figure 6a and Figure 6b), leading to over-detection of transitions. Butterworth filtering (Figure 6c) both removes the artifacts of noise and captures the rapid state transition, even if slightly late.

## 5 Conclusion

When Buridan’s ass is modeled as a dynamical system with a discrete-time, discrete-state Markov process, there are several effective techniques of parameter estimation. These techniques include method of moments type estimators, likelihood-based estimators, and state



**Figure 6:** Graphical comparison of denoising techniques applied to the data vector  $\mathbf{x}$

detection. In higher dimensional extensions of this problem, however, the method of moments approach becomes unmanageable. For  $n$  states, there are  $n(n - 1)$  parameters to estimate, and a method of moments approach requires an equal number of invertible, independent statistics. Moreover, deriving expressions for moments such as mean and variance is made difficult by potentially ill-posed partial differential equations. These problematic PDEs also eliminate the possibility of performing simple, likelihood-based estimation.

The state detector is the one method that generalizes to higher dimensions. Furthermore, state detection is possible in any convex geometry. Therefore, we focus on this method to estimate our parameters in the three-state problem. State detection provides us with the best results we hope to achieve from observing an ideal, noiseless system. Unfortunately, an ideal donkey is not a realistic donkey. Adding noise to the system provides a more realistic problem. State detection alone, however, performs poorly in estimating the parameters, as transitions are over-detected in the presence of noise. Combining our state detector with denoising techniques relieves much of this over-detection. Of the denoising

techniques presented, Butterworth filtering and TV lead to the best parameter estimates.

Further research lies in refining the denoising techniques shown, as well as exploring other denoising methods, to produce better estimates of the parameters. Any of the estimation schemes presented could also be used to inform Markov chain Monte Carlo simulations. Provided with good initial estimates, such simulations can improve convergence by searching over narrower parameter regimes.

## Appendices

### A An Alternative Perspective

Instead of a discrete-time Markov process, we can use a continuous-time Poisson process to model the stochastic switching nature of our dynamical systems. This formulation involves parameters dictating the average time observed between transitions, rather than parameters dictating the probability of transitioning in each unit of time. Nevertheless, the task of estimating these new parameters presents the same challenges associated with measurement error as before. An implementation of this model would again assume knowledge of the donkey's position at finitely many time steps.

In this alternative perspective, we assume that transitions (e.g. state 0 to state 1) each describe a Poisson process, so that the elapsed times between transitions follow exponential distributions. For instance, in the case of donkey on a line, a transition to state 0 is followed by some random amount of time  $dt_{01}$  (before a transition to state 1) sampled from an exponential distribution with mean parameter  $\mu_{01}$ . That is, the random variable  $dt_{01}$  has density

$$f_{01}(t) = \begin{cases} \frac{1}{\mu_{01}} e^{-\frac{1}{\mu_{01}}t} & \text{if } t \geq 0 \\ 0 & \text{if } t < 0. \end{cases} \quad (87)$$

Waiting times for transitions from state 1 to state 0 are also governed by an exponential distribution, just with a different mean parameter  $\mu_{10}$ .

Considering donkey in a triangle, there are two possible transitions from each state. Thus, the dynamical system contains six mean parameters to be estimated. Now, transitions away from a state are viewed as two competing Poisson processes. We can consider two  $dt$ 's for the two possible transitions, and the transition with the smaller  $dt$  will occur.

Generalizing this concept and (87) to any number of states, we say that waiting times for transitions from state  $i$  to state  $j$  are taken from the exponential distribution

$$f_{ij}(t) = \begin{cases} \frac{1}{\mu_{ij}} e^{-\frac{1}{\mu_{ij}}t} & \text{if } t \geq 0 \\ 0 & \text{if } t < 0. \end{cases} \quad (88)$$

We will assume  $0 < \mu_{ij} < \infty$  for all  $i$  and  $j$ . From (88) we obtain the cumulative distribution function

$$F_{ij}(t) = \int_{-\infty}^t f_{ij}(s) ds \quad (89)$$

$$= \int_0^t f_{ij}(s) ds \quad (90)$$

$$= \int_0^t \frac{1}{\mu_{ij}} e^{-\frac{1}{\mu_{ij}} s} ds \quad (91)$$

$$= -e^{-\frac{1}{\mu_{ij}} s} \Big|_0^t \quad (92)$$

$$= 1 - e^{-\frac{1}{\mu_{ij}} t}. \quad (93)$$

With probability  $F_{ij}(t)$  (assuming the donkey's only possible transition is to state  $j$ ), the donkey spends less than  $t$  units of time in state  $i$  before making a transition to state  $j$ ; that is,  $dt_{ij} < t$ . Rather,  $dt_{ij} > t$  with probability

$$1 - F_{ij}(t) = e^{-\frac{1}{\mu_{ij}} t}. \quad (94)$$

This observation leads to the following property of the Poisson process: For any  $s, t > 0$ ,

$$\Pr(dt_{ij} > t + s | dt_{ij} > t) = \frac{\Pr(dt_{ij} > t + s)}{\Pr(dt_{ij} > t)} \quad (95)$$

$$= \frac{e^{-\frac{1}{\mu_{ij}}(t+s)}}{e^{-\frac{1}{\mu_{ij}}t}} \quad (96)$$

$$= e^{-\frac{1}{\mu_{ij}}s} \quad (97)$$

$$= \Pr(dt_{ij} > s). \quad (98)$$

Thus, the likelihood of a transition occurring in a given length of time is independent of when the previous transition occurred. This property is the continuous analog of the time independence that characterizes the steps of a discrete Markov process.

## A.1 Expected Waiting Times

As stated before, when more than one type of transition is possible (i.e. the system has more than two states), we must consider multiple  $dt$ 's, each one an independent, random sample from a separate exponential distribution. But suppose we wanted to know how long, on average, the donkey spends in a given state. In other words, we wish to determine the expected waiting time for *any* transition to occur.

Preserving generality, let there be  $n$  states, and let the donkey be in state  $i$ . We are interested in how long we wait before observing the next transition: the time elapsed (since the transition to state  $i$ ) will be denoted  $dt_i$ . This quantity is given by

$$dt_i = \min(dt_{i0}, \dots, dt_{i,i-1}, dt_{i,i+1}, \dots, dt_{i,n-1}). \quad (99)$$

The probability of having had a transition by time  $t$  is

$$\Pr(dt_i < t) = 1 - \Pr(dt_i > t) \quad (100)$$

$$= 1 - \Pr(dt_{i0} > t) \cdots \Pr(dt_{i,i-1} > t) \Pr(dt_{i,i+1} > t) \cdots \Pr(dt_{i,n-1} > t) \quad (101)$$

$$= 1 - \left(e^{-\frac{1}{\mu_{i0}}t}\right) \cdots \left(e^{-\frac{1}{\mu_{i,i-1}}t}\right) \left(e^{-\frac{1}{\mu_{i,i+1}}t}\right) \cdots \left(e^{-\frac{1}{\mu_{i,n-1}}t}\right) \quad (102)$$

$$= 1 - e^{-\left(\frac{1}{\mu_{i0}} + \cdots + \frac{1}{\mu_{i,i-1}} + \frac{1}{\mu_{i,i+1}} + \cdots + \frac{1}{\mu_{i,n-1}}\right)t}. \quad (103)$$

We recognize this as a cumulative distribution function

$$F_i(t) = 1 - e^{-\left(\sum_{j \neq i} \frac{1}{\mu_{ij}}\right)t}. \quad (104)$$

Thus,  $dt_i$  has density

$$f_i(t) = \frac{d}{dt} F_i \quad (105)$$

$$= \sum_{j \neq i} \frac{1}{\mu_{ij}} e^{-\left(\sum_{j \neq i} \frac{1}{\mu_{ij}}\right)t}. \quad (106)$$

This is an exponential distribution with mean parameter

$$\mu_i = \left(\sum_{j \neq i} \frac{1}{\mu_{ij}}\right)^{-1}. \quad (107)$$

That is, the average waiting time to observe a transition away from state  $i$  is  $\mu_i$  units of time. Assuming the donkey's state information is detectable, we can empirically observe  $\mu_i$  and use such a measurement to estimate the various  $\mu_{ij}$ .

## A.2 Transition Probabilities

We next ask what the probability is of observing a particular transition: If the donkey is in state  $i$ , what is the probability  $p_{ij}$  that his next transition is to state  $j$ ? To calculate this probability, we need only to realize that doing so is equivalent to calculating the probability

that  $dt_{ij} > dt_{ik}$  for all  $k \neq i, j$ . Since  $\Pr(dt_{ij} = dt_{ik} \text{ for some } j \neq k) = 0$ , we do not worry about simultaneous transitions. We find

$$\begin{aligned}
p_{ij} = \Pr(dt_{ij} > dt_{ik} \text{ for all } k \neq i, j) &= \int_0^\infty \left[ \Pr(dt_{ij} = t) \Pr(dt_{i0} > t) \cdots \right. \\
&\quad \Pr(dt_{i,i-1} > t) \Pr(dt_{i,i+1} > t) \cdots \\
&\quad \Pr(dt_{i,j-1} > t) \Pr(dt_{i,j+1} > t) \cdots \\
&\quad \left. \Pr(dt_{i,n-1} > t) \right] dt.
\end{aligned} \tag{108}$$

Using (94), the integral in (108) can be evaluated to give

$$\begin{aligned}
p_{ij} &= \int_0^\infty \left[ \left( \frac{1}{\mu_{ij}} e^{-\frac{1}{\mu_{ij}} t} \right) \left( e^{-\frac{1}{\mu_{i0}} t} \right) \cdots \left( e^{-\frac{1}{\mu_{i,i-1}} t} \right) \left( e^{-\frac{1}{\mu_{i,i+1}} t} \right) \cdots \right. \\
&\quad \left. \left( e^{-\frac{1}{\mu_{i,j-1}} t} \right) \left( e^{-\frac{1}{\mu_{i,j+1}} t} \right) \cdots \left( e^{-\frac{1}{\mu_{i,n-1}} t} \right) \right] dt
\end{aligned} \tag{109}$$

$$= \frac{1}{\mu_{ij}} \int_0^\infty e^{-(\sum_{k \neq i} \frac{1}{\mu_{ik}}) t} dt \tag{110}$$

$$= -\frac{1}{\mu_{ij}} \left( \sum_{k \neq i} \frac{1}{\mu_{ik}} \right)^{-1} e^{-(\sum_{k \neq i} \frac{1}{\mu_{ik}}) t} \Big|_0^\infty \tag{111}$$

$$= \frac{\mu_i}{\mu_{ij}}. \tag{112}$$

That is, we expect

$$p_{ij} = \frac{\mu_i}{\mu_{ij}} \tag{113}$$

of the transitions from state  $i$  to be to state  $j$ . Again, assuming the donkey's state information is detectable, we can measure  $p_{ij}$  (very similar to  $\tau_{ij}$  in the Markov chain formulation) and use it to estimate  $\mu_{ij}$ .

### A.3 Parameter Estimation

We have nearly derived an expression to estimate the various  $\mu_{ij}$  parameters. We now simply solve (113) for  $\mu_{ij}$  to yield

$$\mu_{ij} = \frac{\mu_i}{p_{ij}}. \tag{114}$$

Recall that  $\mu_i$  and  $p_{ij}$  are both easily measured given state information, so the empirical  $\mu_{ij}$  can be calculated.

## B Eigenvector of Markov Matrix with Eigenvalue 1

As presented in 2.1.1 and 3.1, the normalized eigenvector associated with eigenvalue 1 of the  $2 \times 2$

$$\begin{bmatrix} 1 - \tau_{01} & \tau_{10} \\ \tau_{01} & 1 - \tau_{10} \end{bmatrix}$$

and the  $3 \times 3$

$$\begin{bmatrix} 1 - \tau_{01} - \tau_{02} & \tau_{10} & \tau_{20} \\ \tau_{01} & 1 - \tau_{10} - \tau_{12} & \tau_{21} \\ \tau_{02} & \tau_{12} & 1 - \tau_{20} - \tau_{21} \end{bmatrix}$$

Markov matrices are known. In fact, they can be calculated using standard techniques. With increasing size of the Markov matrix, however, these techniques fail by theoretical necessity: Determining the eigenvalues of an arbitrary  $n \times n$  matrix requires solving an  $n$ th degree polynomial, for which a general solution cannot exist for  $n \geq 5$  due to the Abel-Ruffini theorem (Rotman, 2005).

We are only interested in determining the eigenvector associated with eigenvalue 1. Here we derive a formula to calculate this eigenvector given an  $n \times n$  Markov matrix. The following theorem and proof are taken from Lavallée (2011). The result is shown for a general column stochastic matrix.

**Theorem 1.** *If  $A$  is an  $n \times n$  column stochastic matrix, then  $A$  has an eigenvalue of 1 and associated eigenvector  $\mathbf{v}$  given by*

$$v_k = \det(M_{k,k}), \quad (115)$$

where  $M = A - I$  and  $M_{k,k}$  is the matrix formed from  $M$  by removing its  $k$ th row and its  $k$ th column.

*Proof.* We wish to show

$$A\mathbf{v} = \mathbf{v} \quad (116)$$

$$(A - I)\mathbf{v} = \mathbf{0} \quad (117)$$

$$M\mathbf{v} = \mathbf{0}, \quad (118)$$

or equivalently

$$\sum_{j=1}^n m_{i,j} v_j = 0 \quad (119)$$

$$\sum_{j=1}^n m_{i,j} \det M_{j,j} = 0 \quad (120)$$



for  $i = 1, \dots, n$ , where  $m_{i,j}$  is the entry of  $M$  in the  $i$ th row and the  $j$ th column. We will show (120) only for  $i = 1$ , as the remaining  $n - 1$  cases are analogous.

First notice that the rows of  $M$  are linearly dependent (each column sums to 0, so any row is the negative of the sum of all other rows), implying  $\det M = 0$ . We also have

$$\sum_{j=1}^n m_{1,j} \det M_{j,j} = m_{1,1} \begin{vmatrix} m_{2,2} & \cdots & m_{2,n} \\ \vdots & \ddots & \vdots \\ m_{n,2} & \cdots & m_{n,n} \end{vmatrix} + \cdots + m_{1,n} \begin{vmatrix} m_{1,1} & \cdots & m_{1,n-1} \\ \vdots & \ddots & \vdots \\ m_{n-1,1} & \cdots & m_{n-1,n-1} \end{vmatrix} \quad (121)$$

$$\begin{aligned} &= m_{1,1} \begin{vmatrix} m_{2,2} & \cdots & m_{2,n} \\ \vdots & \ddots & \vdots \\ m_{n,2} & \cdots & m_{n,n} \end{vmatrix} + m_{1,2} \begin{vmatrix} -\sum_{j=2}^n m_{j,1} & -\sum_{j=2}^n m_{j,3} & \cdots & -\sum_{j=2}^n m_{j,n} \\ m_{3,1} & m_{3,3} & \cdots & m_{3,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,3} & \cdots & m_{n,n} \end{vmatrix} \\ &+ \cdots + m_{1,n} \begin{vmatrix} -\sum_{j=2}^n m_{j,1} & -\sum_{j=2}^n m_{j,2} & \cdots & -\sum_{j=2}^n m_{j,n-1} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n-1,1} & m_{n-1,2} & \cdots & m_{n-1,n-1} \end{vmatrix}, \end{aligned} \quad (122)$$

where, in the last  $n - 1$  terms of the sum, we have expressed the first row of the matrix as the negative of the sum of the last  $n - 1$  rows (with the appropriate columns omitted) of  $M$ . In computing the determinants of these matrices, we now remove the negative from the first row and split the sums:

$$\begin{aligned} \sum_{j=1}^n m_{1,j} \det M_{j,j} &= m_{1,1} \begin{vmatrix} m_{2,2} & \cdots & m_{2,n} \\ \vdots & \ddots & \vdots \\ m_{n,2} & \cdots & m_{n,n} \end{vmatrix} - m_{1,2} \sum_{j=2}^n \begin{vmatrix} m_{j,1} & m_{j,3} & \cdots & m_{j,n} \\ m_{3,1} & m_{3,3} & \cdots & m_{3,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,3} & \cdots & m_{n,n} \end{vmatrix} \\ &\quad - \cdots - m_{1,n} \sum_{j=2}^n \begin{vmatrix} m_{j,1} & m_{j,2} & \cdots & m_{j,n-1} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n-1,1} & m_{n-1,2} & \cdots & m_{n-1,n-1} \end{vmatrix}. \end{aligned} \quad (123)$$

In each of the  $n - 1$  sums on the right-hand side of (123), only one  $j$  yields a first row that is not identical to one of the other rows (in particular, for the sum following  $m_{1,k}$ , this  $j$

equals  $k$ ). As all other  $j$  yield a determinant of 0, each sum reduces to a single term:

$$\begin{aligned} \sum_{j=1}^n m_{1,j} \det M_{j,j} &= m_{1,1} \begin{vmatrix} m_{2,2} & \cdots & m_{2,n} \\ \vdots & \ddots & \vdots \\ m_{n,2} & \cdots & m_{n,n} \end{vmatrix} - m_{1,2} \begin{vmatrix} m_{2,1} & m_{2,3} & \cdots & m_{2,n} \\ m_{3,1} & m_{3,3} & \cdots & m_{3,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,3} & \cdots & m_{n,n} \end{vmatrix} \\ &\quad - \cdots - m_{1,n} \begin{vmatrix} m_{n,1} & m_{n,2} & \cdots & m_{n,n-1} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n-1,1} & m_{n-1,2} & \cdots & m_{n-1,n-1} \end{vmatrix}. \end{aligned} \quad (124)$$

We now shift down the first row of the last  $n-2$  matrices in (124) so that rows are in order (for the  $m_{1,k}$  term, this requires  $k-2$  transpositions, or “flips,” of rows). Furthermore, each transposition changes the sign of the determinant, so we have

$$\begin{aligned} \sum_{j=1}^n m_{1,j} \det M_{j,j} &= m_{1,1} \begin{vmatrix} m_{2,2} & \cdots & m_{2,n} \\ \vdots & \ddots & \vdots \\ m_{n,2} & \cdots & m_{n,n} \end{vmatrix} + (-1)^1 m_{1,2} \begin{vmatrix} m_{2,1} & m_{2,3} & \cdots & m_{2,n} \\ m_{3,1} & m_{3,3} & \cdots & m_{3,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,3} & \cdots & m_{n,n} \end{vmatrix} \\ &\quad + \cdots + (-1)^{n-1} m_{1,n} \begin{vmatrix} m_{2,1} & m_{2,2} & \cdots & m_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n-1,1} & m_{n-1,2} & \cdots & m_{n-1,n-1} \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n-1} \end{vmatrix} \end{aligned} \quad (125)$$

$$\begin{aligned} &= (-1)^{1+1} m_{1,1} \begin{vmatrix} m_{2,2} & \cdots & m_{2,n} \\ \vdots & \ddots & \vdots \\ m_{n,2} & \cdots & m_{n,n} \end{vmatrix} \\ &\quad + (-1)^{1+2} m_{1,2} \begin{vmatrix} m_{2,1} & m_{2,3} & \cdots & m_{2,n} \\ m_{3,1} & m_{3,3} & \cdots & m_{3,n} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n,1} & m_{n,3} & \cdots & m_{n,n} \end{vmatrix} \\ &\quad + \cdots + (-1)^{1+n} m_{1,n} \begin{vmatrix} m_{2,1} & m_{2,2} & \cdots & m_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n-1,1} & m_{n-1,2} & \cdots & m_{n-1,n-1} \\ m_{n,1} & m_{n,2} & \cdots & m_{n,n-1} \end{vmatrix} \end{aligned} \quad (126)$$

$$\begin{aligned} &= \sum_{j=1}^n (-1)^{1+j} m_{1,j} \det M_{1,j}. \end{aligned} \quad (127)$$

We recognize (127) as Laplace's formula by cofactor expansion for the determinant (Treil, 2009), so we have

$$\sum_{j=1}^n m_{1,j} \det M_{1,j} = \sum_{j=1}^n (-1)^{1+j} m_{1,j} \det M_{1,j} \quad (128)$$

$$= \det M \quad (129)$$

$$= 0. \quad (130)$$

□

## C A Conjecture Regarding Eigenvectors

Theorem 1 provides a way of computing the eigenvector corresponding to the eigenvalue 1 of the  $n \times n$  Markov matrix

$$A = \begin{bmatrix} 1 - \tau_{01} - \cdots - \tau_{0,n-1} & \tau_{10} & \cdots & \tau_{n-1,0} \\ \tau_{01} & 1 - \tau_{10} - \tau_{12} - \cdots - \tau_{1,n-1} & \cdots & \tau_{n-1,1} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{0,n-1} & \tau_{1,n-1} & \cdots & 1 - \tau_{n-1,0} - \cdots - \tau_{n-1,n-2} \end{bmatrix}. \quad (131)$$

Nevertheless, the computation (115) has increasing complexity with increasing  $n$ . For example, using MATLAB<sup>®</sup>, we were able to compute the desired eigenvector only for  $n \leq 6$ . For  $n > 6$ , the computation proved too complex to execute. To gain a sense of this complexity, we examined the growth of the number of summands in a single application of (115).

For example, in the  $2 \times 2$  case,  $A$  has the normalized eigenvector

$$\mathbf{v} = \begin{bmatrix} \frac{\tau_{10}}{\tau_{01} + \tau_{10}} \\ \frac{\tau_{01}}{\tau_{01} + \tau_{10}} \end{bmatrix}.$$

In the  $3 \times 3$  case,

$$\mathbf{v} = \begin{bmatrix} \frac{\tau_{21}\tau_{10} + \tau_{12}\tau_{20} + \tau_{10}\tau_{20}}{\tau_{21}\tau_{10} + \tau_{12}\tau_{20} + \tau_{10}\tau_{20} + \tau_{20}\tau_{01} + \tau_{02}\tau_{21} + \tau_{01}\tau_{21} + \tau_{10}\tau_{02} + \tau_{01}\tau_{12} + \tau_{02}\tau_{12}} \\ \frac{\tau_{20}\tau_{01} + \tau_{02}\tau_{21} + \tau_{01}\tau_{21}}{\tau_{21}\tau_{10} + \tau_{12}\tau_{20} + \tau_{10}\tau_{20} + \tau_{20}\tau_{01} + \tau_{02}\tau_{21} + \tau_{01}\tau_{21} + \tau_{10}\tau_{02} + \tau_{01}\tau_{12} + \tau_{02}\tau_{12}} \\ \frac{\tau_{10}\tau_{02} + \tau_{01}\tau_{12} + \tau_{02}\tau_{12}}{\tau_{21}\tau_{10} + \tau_{12}\tau_{20} + \tau_{10}\tau_{20} + \tau_{20}\tau_{01} + \tau_{02}\tau_{21} + \tau_{01}\tau_{21} + \tau_{10}\tau_{02} + \tau_{01}\tau_{12} + \tau_{02}\tau_{12}} \end{bmatrix}.$$

With four states, the eigenvector (not normalized) becomes

$$\mathbf{v} = \begin{bmatrix} \tau_{10}\tau_{20}\tau_{30} + \tau_{10}\tau_{20}\tau_{31} + \tau_{10}\tau_{21}\tau_{30} + \tau_{10}\tau_{20}\tau_{32} + \tau_{10}\tau_{21}\tau_{31} + \tau_{12}\tau_{20}\tau_{30} + \tau_{10}\tau_{21}\tau_{32} + \tau_{10}\tau_{23}\tau_{30} \\ + \tau_{12}\tau_{20}\tau_{31} + \tau_{13}\tau_{20}\tau_{30} + \tau_{10}\tau_{23}\tau_{31} + \tau_{12}\tau_{20}\tau_{32} + \tau_{13}\tau_{21}\tau_{30} + \tau_{12}\tau_{23}\tau_{30} + \tau_{13}\tau_{20}\tau_{32} + \tau_{13}\tau_{23}\tau_{30} \\ \tau_{01}\tau_{20}\tau_{30} + \tau_{01}\tau_{20}\tau_{31} + \tau_{01}\tau_{21}\tau_{30} + \tau_{01}\tau_{20}\tau_{32} + \tau_{01}\tau_{21}\tau_{31} + \tau_{02}\tau_{21}\tau_{30} + \tau_{01}\tau_{21}\tau_{32} + \tau_{01}\tau_{23}\tau_{30} \\ + \tau_{02}\tau_{21}\tau_{31} + \tau_{03}\tau_{20}\tau_{31} + \tau_{01}\tau_{23}\tau_{31} + \tau_{02}\tau_{21}\tau_{32} + \tau_{03}\tau_{21}\tau_{31} + \tau_{02}\tau_{23}\tau_{31} + \tau_{03}\tau_{21}\tau_{32} + \tau_{03}\tau_{23}\tau_{31} \\ \tau_{02}\tau_{10}\tau_{30} + \tau_{01}\tau_{12}\tau_{30} + \tau_{02}\tau_{10}\tau_{31} + \tau_{01}\tau_{12}\tau_{31} + \tau_{02}\tau_{10}\tau_{32} + \tau_{02}\tau_{12}\tau_{30} + \tau_{01}\tau_{12}\tau_{32} + \tau_{02}\tau_{12}\tau_{31} \\ + \tau_{02}\tau_{13}\tau_{30} + \tau_{03}\tau_{10}\tau_{32} + \tau_{01}\tau_{13}\tau_{32} + \tau_{02}\tau_{12}\tau_{32} + \tau_{03}\tau_{12}\tau_{31} + \tau_{02}\tau_{13}\tau_{32} + \tau_{03}\tau_{12}\tau_{32} + \tau_{03}\tau_{13}\tau_{32} \\ \tau_{03}\tau_{10}\tau_{20} + \tau_{01}\tau_{13}\tau_{20} + \tau_{03}\tau_{10}\tau_{21} + \tau_{01}\tau_{13}\tau_{21} + \tau_{02}\tau_{10}\tau_{23} + \tau_{03}\tau_{12}\tau_{20} + \tau_{01}\tau_{12}\tau_{23} + \tau_{02}\tau_{13}\tau_{21} \\ + \tau_{03}\tau_{10}\tau_{23} + \tau_{03}\tau_{13}\tau_{20} + \tau_{01}\tau_{13}\tau_{23} + \tau_{02}\tau_{12}\tau_{23} + \tau_{03}\tau_{13}\tau_{21} + \tau_{02}\tau_{13}\tau_{23} + \tau_{03}\tau_{12}\tau_{23} + \tau_{03}\tau_{13}\tau_{23} \end{bmatrix}.$$

It is easy to see from (115) that for a given  $n$ , each entry of the eigenvector  $\mathbf{v}$  will contain the same number of summands before normalizing. Counting the number of monomials in each numerator after normalizing, we see a possible pattern, illustrated in Table 8.

**Table 8:** Number of monomials in a coordinate of the eigenvector corresponding to eigenvalue 1 of Markov matrix

Number of States	Terms in Numerators	
2	1	$= 2^0$
3	3	$= 3^1$
4	16	$= 4^2$
5	125	$= 5^3$
6	1296	$= 6^4$

We are lead to the following conjecture.

**Conjecture 1.** *If  $A$  is the  $n \times n$  matrix given by (131), then each coordinate of the eigenvector  $\mathbf{v}$  given by (115), when expressed in terms of the various  $\tau_{ij}$ , is the sum of  $n^{n-2}$  monomials, each with coefficient 1.*

If this conjecture is true, a derivation may proceed as follows. We let  $\mathcal{R}$  be the number of monomials in a coordinate of  $\mathbf{v}$ , the number we conjecture is equal to  $n^{n-2}$ . Let  $M = A - I$  as in Appendix B. As prescribed by Theorem 1, to compute the  $k$ th entry of  $\mathbf{v}$ , we must calculate  $\det M_{k,k}$ . To do so, we will use the Leibniz formula (Treil, 2009), presented here for a general matrix  $B$ :

$$\det B = \sum_{\sigma \in S_n} \text{sgn } \sigma \prod_{i=1}^n b_{i,\sigma(i)}, \quad (132)$$

where  $S_n$  is the symmetric group on  $n$  elements, and  $b_{i,j}$  denotes an entry in  $B$ .

In applying this formula, for example, to

$$M_{n,n} = \begin{bmatrix} -\tau_{01} - \cdots - \tau_{0,n-1} & \tau_{10} & \cdots & \tau_{n-2,0} \\ \tau_{01} & -\tau_{10} - \tau_{12} - \cdots - \tau_{1,n-1} & \cdots & \tau_{n-2,1} \\ \vdots & \vdots & \ddots & \vdots \\ \tau_{0,n-2} & \tau_{1,n-2} & \cdots & -\tau_{n-2,0} - \cdots - \tau_{n-2,n-3} - \tau_{n-2,n-1} \end{bmatrix},$$

we notice that

1. Each term in the sum of (132) is a product of  $n - 1$  entries of  $M_{n,n}$ , exactly one from each row and column.
2. In a particular summand of (132), the sign of each monomial produced is the same and is determined by two things. First, each diagonal entry in the product forming the monomial contributes a power of  $-1$ . Second, the signum of the permutation involved could contribute an additional factor of  $-1$ .
3. Diagonal entries being involved in a particular summand of (132) is equivalent to the permutation involved having a fixed point.
4. The number of diagonal entries involved in a particular summand of (132) determines how many monomials the summand contains. In particular, if  $f(\sigma)$  denotes the number of fixed points of the permutation  $\sigma$ , then the associated summand will contain  $(n - 1)^{f(\sigma)}$  monomials.

The assumption we now make, albeit possibly incorrect, is that monomials of opposite sign will cancel to leave only monomials with identical sign. In this case, the four observations above imply

$$\mathcal{R} = (-1)^{n-1} \sum_{\sigma \in S_n} (\text{sgn } \sigma) (-1)^{f(\sigma)} (n - 1)^{f(\sigma)}. \quad (133)$$

The factor of  $(-1)^{n-1}$  in (133) ensures  $\mathcal{R}$  is always positive. If  $n$  is even, then the monomials comprising  $\mathbf{v}$ , as determined by (115), have negative coefficients. This fact is seen by noting that the majority of monomials come from the summand associated with the identity permutation—the summand in (132) that multiplies all diagonal elements. Since the identity permutation has positive signum, the sign of the monomials in this summand is entirely determined by  $n$ . If  $n$  is even, then  $n - 1$  is odd, so the sign of these monomials is  $(-1)^{n-1}$ , as there are  $n - 1$  diagonal entries.

Provided our assumption is correct, proving Conjecture 1 is now reduced to showing the right-hand side of (133) is equal to  $n^{n-2}$ . The sum in (133) may be easier to manipulate if permutations were grouped by cycle structure.

## D The Growth of Cumulative Power

In Quinn (2011), the author shows that, for finite sums of sines and cosines, cumulative power (28) is  $\mathcal{O}(t)$ . Here we prove the same growth relationship for the cumulative power (29) of the donkey's position,  $x$ , in the one-dimensional case.

**Theorem 2.** *If  $x$  obeys (1) and (2) and switches between the two states by the Markov process (3), then its cumulative power (29) is  $\mathcal{O}(t)$ .*

*Proof.* Notice from (34) and (35) that

$$0 < (x''(s))^2 < v^4, \quad (134)$$

since

$$0 < x(s) < 1 \quad (135)$$

for all time  $s$ , under the assumption  $0 < x(0) < 1$ . Thus, we have

$$F(t) = \int_0^{t_1} (x''(s))^2 ds + \int_{t_1}^{t_2} (x''(s))^2 ds + \cdots + \int_{t_{n-1}}^{t_n} (x''(s))^2 ds + \int_{t_n}^t (x''(s))^2 ds \quad (136)$$

$$< \int_0^{t_1} v^4 ds + \int_{t_1}^{t_2} v^4 ds + \cdots + \int_{t_{n-1}}^{t_n} v^4 ds + \int_{t_n}^t v^4 ds \quad (137)$$

$$= v^4 [(t_1 - 0) + (t_2 - t_1) + \cdots + (t_n - t_{n-1}) + (t - t_n)] \quad (138)$$

$$= v^4 t, \quad (139)$$

for all time  $t$ . Since  $v^4$  is constant and cumulative power  $F$  is always positive, (139) shows  $F \in \mathcal{O}(t)$ .  $\square$

## Acknowledgments

This work was completed as part of the Summer Undergraduate Research Institute in Experimental Mathematics (SURIEM) at the Lyman Briggs College of Michigan State University. We thank the sponsorship of the National Security Agency and the National Science Foundation in funding this REU program. We thank Professor Daniel P. Dougherty for his mentorship, as well as Joseph E. Roth for his assistance. We also wish to credit Mr. Roth for his observations leading to Conjecture 1.

## References

Abramowitz M. and Stegun I. 1964. Handbook of mathematical functions with formulas, graphs, and mathematical tables. Dover publications.

- Butterworth S. 1930. On the theory of filter amplifiers. *Wireless Engineer* 7:536–541.
- Cleveland W.S. 1979. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Association* 74, 368:829–836.
- Fan J. and Gijbels I. 1996. *Local polynomial modelling and its applications*. Chapman and Hall.
- Goldstein T. and Osher S. 2009. The split bregman method for l1 regularized problems. *SIAM Journal on Imaging Sciences* 2, 2:323–343.
- Hansen L. 1982. Large sample properties of generalized method of moments estimators. *Econometrica: Journal of the Econometric Society* pp. 1029–1054.
- Johnson N., Kotz S., and Balakrishnan N. 1995. Continuous univariate distributions. Number v. 2 in *Wiley series in probability and mathematical statistics: Applied probability and statistics*. Wiley & Sons. ISBN 9780471584940.
- Lamport L. 2012. Buridan’s principle. *Foundations of Physics* pp. 1–11.
- Lavallée S. 2011. Left eigenvector of a stochastic matrix. *Advances in Pure Mathematics* 1, 4:105–117.
- Mallat S. 2008. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition. ISBN 9780123743701.
- Orey S. 1962. An ergodic theorem for markov chains. *Probability Theory and Related Fields* 1, 2:174–176.
- Polyanin A. and Zaitsev V. 2004. *Handbook of nonlinear partial differential equations*. CRC press.
- Quinn K. 2011. Inverse modeling by cumulative-power-penalized least squares.
- Rotman J. 2005. *A First Course in Abstract Algebra*. Prentice Hall, 3rd edition. ISBN 9780131862678.
- Rudin L., Osher S., and Fatemi E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* 60, 1-4:259–268.
- Runge C. 1901. Über empirische funktionen und die interpolation zwischen äquidistanten ordinaten. *Zeitschrift für Mathematik und Physik* 46, 224-243:20.
- Treil S. 2009. *Linear algebra done wrong*.